*Acta Numerica* (2018), pp. 207–287
doi:10.1017/S0962492917000113

© Cambridge University Press, 2018
Printed in the United Kingdom

# Numerical methods for nonlinear equations

C. T. Kelley
*Department of Mathematics,*
*North Carolina State University,*
*Raleigh, NC 27695-8205, USA*
*E-mail:* tim_kelley@ncsu.edu

This article is about numerical methods for the solution of nonlinear equations. We consider both the fixed-point form $\mathbf{x} = \mathbf{G}(\mathbf{x})$ and the equations form $\mathbf{F}(\mathbf{x}) = 0$ and explain why both versions are necessary to understand the solvers. We include the classical methods to make the presentation complete and discuss less familiar topics such as Anderson acceleration, semi-smooth Newton's method, and pseudo-arclength and pseudo-transient continuation methods.

## CONTENTS

1 Introduction 207
2 Newton's method: classical algorithms 210
3 The Kantorovich theorem and mesh independence 237
4 Pseudo-arclength continuation 245
5 Anderson acceleration 254
6 Newton's method for semi-smooth functions 265
7 Pseudo-transient continuation 272
References 279

## 1. Introduction

Nonlinear equations are ubiquitous, and methods for their solution date from the quadratic formula. Modern numerical methods are, for the most part, based on Newton's method or Picard iteration.

Most of the results in this paper, while stated and proved in a finite-dimensional setting, do not depend on compactness of the unit ball and are valid in a Banach space setting (see Section 2.9). We will explicitly point out the few exceptions.

This subject is old (Newton 1967–1976, Raphson 1690, Picard 1890). Ortega and Rheinboldt (1970), Dennis and Schnabel (1996) and Kelley (1995) are our primary sources for notation and analysis of the classical methods. The bibliographies of these books vividly illustrate the rich classical literature in this field. The approach we take in this article is not the only one. Deuflhard (2004), for example, has a somewhat different viewpoint.

We intend this article to be self-contained for any student of numerical analysis. To that end we summarize the classical theory of Newton and Newton-iterative methods in Section 2. The remaining sections have less familiar material and could in many ways be thought of as a second volume of Kelley (1995). The topics in Sections 5 and 6, in particular, are very active areas of research.

### *1.1. Notation*

We seek to solve nonlinear equations in $\mathbb{R}^N$. We will write vectors in boldface lower-case, maps on vectors in boldface upper-case, and components of vectors as lower-case roman letters. For example, if $\mathbf{x} \in \mathbb{R}^N$, $x_i$ is the $i$th component of $\mathbf{x}$. The methods are iterative, and we will denote the sequence of iterations by $\{\mathbf{x}_n\}$ when the entire sequence (or several elements of the sequence) is of interest. In many cases only the current iteration $\mathbf{x}_c$ and the next one $\mathbf{x}_+$ are needed, and we can express the algorithm in terms of the transition from $\mathbf{x}_c$ to $\mathbf{x}_+$.

Two formulations of nonlinear equations are of interest in this article.

### *1.2. Root finding formulation and Newton's method*

The 'root finding' form is

$$\mathbf{F}(\mathbf{x}) = 0, \tag{1.1}$$

where $\mathbf{F} : \mathbb{R}^N \to \mathbb{R}^N$,

$$\mathbf{F}(\mathbf{x}) = \begin{pmatrix} f_1(\mathbf{x}) \\ f_2(\mathbf{x}) \\ \vdots \\ f_N(\mathbf{x}) \end{pmatrix}.$$

We will refer to $\mathbf{F}$ as the *residual.*

If $\mathbf{F}$ is differentiable at $\mathbf{x}$, we will denote the Jacobian matrix by $\mathbf{F}'(\mathbf{x})$. Recall that

$$\mathbf{F}'(x)_{ij} = \partial f_i(x)/\partial x_j.$$

When we express an equation in this form we will be solving it with a variation of Newton's method. The classical version of Newton's method takes $\mathbf{x}_c$ to $\mathbf{x}_+$ via

$$\mathbf{x}_+ = \mathbf{x}_c - \mathbf{F}'(\mathbf{x}_c)^{-1}\mathbf{F}(\mathbf{x}_c). \tag{1.2}$$

Implicit in (1.2) is the solution of the linearized equation for the step $\mathbf{s}$:

$$\mathbf{F}'(\mathbf{x}_c)\mathbf{s} = -\mathbf{F}(\mathbf{x}_c). \tag{1.3}$$

The various formulations of Newton's method we consider in Section 2 differ in the way they approximate a solution to (1.3). In Section 6 we show how to relax the smoothness assumptions on $\mathbf{F}$.

### *1.3. Fixed-point formulation and Picard iteration*

The fixed-point formulation of a nonlinear equation is

$$\mathbf{x} = \mathbf{G}(\mathbf{x}), \tag{1.4}$$

where $\mathbf{G} : \mathbb{R}^N \to \mathbb{R}^N$. The two formulations are equivalent via $\mathbf{F}(\mathbf{x}) = \mathbf{x} - \mathbf{G}(\mathbf{x})$, of course, but the choice of formulation usually carries meaning. In particular, the use of the fixed-point formulation will imply, at least in this article, that derivative information is either not necessary or difficult to obtain when designing the algorithms. Picard iteration (Picard 1890),

$$\mathbf{x}_+ = \mathbf{G}(\mathbf{x}_c),$$

is the classic example of a method that does not use Jacobian information. Picard iteration is also called fixed-point iteration, Richardson iteration or successive substitution. Ortega and Rheinboldt (1970) make a distinction between some of these terms, but we see no reason for that, and our usage reflects common practice. Tapia, Dennis and Schäfermeyer (2018) have an interesting historical perspective.

We will solve fixed-point problems with Picard iteration or one of its variations.

We close this section with the well-known theory for Picard iteration.

**Definition 1.1.** A map $\mathbf{G}$ is a contraction on a closed set $\mathcal{D} \subset \mathbb{R}^N$ if

- $\mathbf{G}(\mathbf{x}) \in \mathcal{D}$ if $\mathbf{x} \in \mathcal{D}$,
- there is $\alpha \in (0, 1)$ such that

  $$\|\mathbf{G}(\mathbf{x}) - \mathbf{G}(\mathbf{y})\| \leq \alpha\|\mathbf{x} - \mathbf{y}\|,$$

  for all $\mathbf{x}, \mathbf{y} \in \mathcal{D}$.

The convergence result is the *contraction mapping theorem.*

**Theorem 1.2.** If $\mathbf{G}$ is a contraction on a closed set $\mathcal{D} \subset \mathbb{R}^N$, then

- there is a unique solution $\mathbf{x}^* \in \mathcal{D}$ to $\mathbf{x} = \mathbf{G}(\mathbf{x})$,
- if $\mathbf{x}_0 \in \mathcal{D}$ then the Picard iteration converges to $\mathbf{x}^*$.

We refer to Ortega and Rheinboldt (1970) and Kelley (1995) for the familiar proof. We will discuss an important extension of Picard iteration in Section 5.

## 2. Newton's method: classical algorithms

For most of this section we will make the so-called standard assumptions on the nonlinear equation $\mathbf{F}(\mathbf{x}) = 0$.

**Assumption 2.1.** There are $\mathbf{x}^* \in \mathbb{R}^N$ and $\rho^* > 0$ such that

- $\mathbf{F}(\mathbf{x}^*) = 0$,
- $\mathbf{F}'(\mathbf{x}^*)$ is non-singular, and
- $\mathbf{F}'(\mathbf{x})$ is Lipschitz continuous with Lipschitz constant $\gamma$, that is,

$$\|\mathbf{F}'(\mathbf{x}) - \mathbf{F}'(\mathbf{y})\| \leq \gamma \|\mathbf{x} - \mathbf{y}\|, \tag{2.1}$$

for all

$$\mathbf{x}, \mathbf{y} \in \mathcal{B}(\mathbf{x}^*, \rho^*) \equiv \{\mathbf{z} \mid \|\mathbf{z} - \mathbf{x}^*\| \leq \rho^*\}.$$

The standard assumptions distinguish the root $\mathbf{x}^*$ of $\mathbf{F}$ from any others and the local convergence theory refers only to that root. When non-uniqueness is an issue (see Sections 4 and 7), then the standard assumptions only play a role after a particular root has been identified.

In this section we will analyse the convergence of Newton's method twice: once for the simple formulation and again to account for errors in the evaluation of $\mathbf{F}$ and in the solution of the linearized problem for the step. The latter of the two results, Theorem 2.3, serves to explain not only many classical variations of Newton's method but also the modern Jacobian-free Newton–Krylov (JFNK) methods (Knoll and Keyes 2004) that are the basis of large-scale nonlinear solvers such as KINSOL (Collier, Hindmarsh, Serban and Woodward 2015), NOX (Heroux *et al.* 2005) and SNES (Balay *et al.* 2015).

### *2.1. Local convergence of Newton's method*

The reader may know Theorem 2.2 well. The simple statement is that if the standard assumptions hold and the initial iterate[1] is sufficiently near $\mathbf{x}^*$ (hence the term *local*), then the Newton iteration $\{\mathbf{x}_n\}$ exists (*i.e.* $\mathbf{F}'(\mathbf{x}_n)$ is non-singular for all $n \geq 0$) and converges quadratically to $\mathbf{x}^*$. The Newton iterates are, of course,

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \mathbf{F}(\mathbf{x}_n)^{-1}\mathbf{F}(\mathbf{x}_n),$$

for $n \geq 0$. This is exactly (1.2) with $\mathbf{x}_c$ replaced by $\mathbf{x}_n$ and $\mathbf{x}_+$ by $\mathbf{x}_{n+1}$. The advantage of the $\mathbf{x}_c$ and $\mathbf{x}_+$ notation is that the transition from $\mathbf{x}_c$ to $\mathbf{x}_+$ is central and the iteration counter is generally irrelevant to the convergence analysis. Quadratic convergence means that

$$\|\mathbf{e}_+\| = O(\|\mathbf{e}_c\|^2), \tag{2.2}$$

---

[1] Not guess! We are professionals here.

where the error in $\mathbf{x}$ is $\mathbf{e} = \mathbf{x} - \mathbf{x}^*$. Quadratic convergence says that the number of significant figures in the result roughly doubles with each iteration.

We will begin with a precise statement of this result which will, among other things, exhibit the constant in the $O$-term. We will also begin to develop a taxonomy of convergence types. We say that $\mathbf{x}_n \to \mathbf{x}^*$ *q-quadratically* if (2.2) holds and *r-quadratically* if there is a real sequence $\{\xi_n\}$ which converges q-quadratically to 0 such that $\|\mathbf{e}_n\| \leq \xi_n$. We say the convergence is *q-linear* if there is $\alpha \in [0,1)$ such that $\|\mathbf{e}_{n+1}\| \leq \alpha \|\mathbf{e}_n\|$ for $n$ sufficiently large; $\alpha$ is called the *q-factor*. The convergence is *q-superlinear* if

$$\lim_{n\to\infty} \frac{\|\mathbf{e}_{n+1}\|}{\|\mathbf{e}_n\|} = 0.$$

Finally, we will quantify 'sufficiently near $\mathbf{x}^*$'. At a minimum, all the local convergence results in this section require

$$\|\mathbf{x} - \mathbf{x}^*\| \leq \min\left( \frac{\|\mathbf{F}'(\mathbf{x}^*)^{-1}\|^{-1}}{2\gamma}, \rho^* \right), \tag{2.3}$$

where $\gamma$ is the Lipschitz constant for $\mathbf{F}'$ and $\rho^*$ is the radius of the ball about $\mathbf{x}^*$ in which the Lipschitz continuity assumption on $\mathbf{F}'$ holds.

### 2.2. *Classical Newton's method*

We will prove Theorem 2.2 in detail. Not only is the proof illuminating in its own right, but some of the details lead to algorithmic insights.

**Theorem 2.2.** Let Assumption 2.1 hold and assume that $\mathbf{x}_c$ satisfies (2.3). Then $\mathbf{F}(\mathbf{x}_c)$ is non-singular,

$$\|\mathbf{F}'(\mathbf{x}_c)\|^{-1} \leq 2\|\mathbf{F}'(\mathbf{x}^*)\|^{-1}, \tag{2.4}$$

and

$$\|\mathbf{e}_+\| \leq \|\mathbf{F}'(\mathbf{x}^*)^{-1}\|\gamma\, \|\mathbf{e}_c\|^2 \leq \|\mathbf{e}_c\|/2. \tag{2.5}$$

*Proof.* The standard assumptions and (2.3) imply that

$$\|\mathbf{F}'(\mathbf{x}_c) - \mathbf{F}'(\mathbf{x}^*)\| \leq \gamma\|\mathbf{e}_c\| \leq \frac{\|\mathbf{F}'(\mathbf{x}^*)^{-1}\|}{2},$$

and hence

$$\|\mathbf{I} - \mathbf{F}'(\mathbf{x}^*)^{-1}\mathbf{F}'(\mathbf{x}_c)\| \leq \|\mathbf{F}'(\mathbf{x}^*)^{-1}\|\|\mathbf{F}'(\mathbf{x}_c) - \mathbf{F}'(\mathbf{x}^*)\| \leq 1/2,$$

so $\mathbf{F}'(\mathbf{x}^*)^{-1}$ is an approximate inverse of $\mathbf{F}'(\mathbf{x}_c)$, and

$$\|\mathbf{F}'(\mathbf{x}_c)^{-1}\| \leq \frac{\|\mathbf{F}'(\mathbf{x}^*)^{-1}\|}{2},$$

as asserted.

The remainder of the proof follows from the fundamental theorem of calculus, which implies that

$$\begin{aligned}\mathbf{F}(\mathbf{x}_c) &= \int_0^1 \mathbf{F}'(\mathbf{x}^* + t\mathbf{e}_c)\mathbf{e}_c\,\mathrm{d}t \\ &= \mathbf{F}'(\mathbf{x}_c)\mathbf{e}_c + \int_0^1 (\mathbf{F}'(\mathbf{x}^* + t\mathbf{e}_c) - \mathbf{F}'(\mathbf{x}_c))\mathbf{e}_c\,\mathrm{d}t. \qquad (2.6)\end{aligned}$$

Hence,

$$\begin{aligned}\mathbf{e}_+ &= \mathbf{e}_c - \mathbf{F}'(\mathbf{x}_c)^{-1}\biggl(\mathbf{F}'(\mathbf{x}_c)\mathbf{e}_c + \int_0^1 (\mathbf{F}'(\mathbf{x}^* + t\mathbf{e}_c) - \mathbf{F}'(\mathbf{x}_c))\mathbf{e}_c\,\mathrm{d}t\biggr) \\ &= -\mathbf{F}'(\mathbf{x}_c)^{-1}\biggl(\int_0^1 (\mathbf{F}'(\mathbf{x}^* + t\mathbf{e}_c) - \mathbf{F}'(\mathbf{x}_c))\mathbf{e}_c\,\mathrm{d}t\biggr).\end{aligned}$$

Note that

$$\biggl\|\int_0^1 (\mathbf{F}'(\mathbf{x}^* + t\mathbf{e}_c) - \mathbf{F}'(\mathbf{x}_c))\mathbf{e}_c\,\mathrm{d}t\biggr\| \leq \int_0^1 \gamma\|\mathbf{e}_c\|^2(1-t)\,\mathrm{d}t = \gamma\|\mathbf{e}_c\|^2/2.$$

Hence,

$$\begin{aligned}\|\mathbf{e}_+\| &\leq \frac{\|\mathbf{F}'(\mathbf{x}_c)^{-1}\|\gamma}{2}\|\mathbf{e}_c\|^2 \leq \frac{2\|\mathbf{F}'(\mathbf{x}^*)^{-1}\|\gamma}{2}\|\mathbf{e}_c\|^2 \\ &\leq \|\mathbf{e}_c\|/2. \qquad (2.7)\end{aligned}$$

This completes the proof and shows that the constant in the $O$-term for quadratic convergence is no larger than $\|\mathbf{F}'(\mathbf{x}^*)^{-1}\|\gamma$. ☐

Our bounds for the distance of the initial iterate from the root and for the convergence rate depend only on the norm of the inverse of the Jacobian at the root and the Lipschitz constant of the Jacobian near the root. This observation leads directly to the Kantorovich theorem (Kantorovich and Akilov 1982: see Section 3) and the implicit function theorem (Keller 1987) in Section 4, which are the basis for the pseudo-arclength continuation method we discuss in Section 4.

### *2.3. Termination criteria*

The only obvious ways to terminate the Newton iteration are to examine the norm of the residual $\|\mathbf{F}(\mathbf{x}_n)\|$ or the norm of the step $\|\mathbf{x}_{n+1} - \mathbf{x}_n\|$. Either way is fine if one computes the Newton step exactly.

The norm of the step is a very good surrogate for the norm of the error at the previous iteration. To see this, note that quadratic convergence implies that

$$\mathbf{x}_{n+1} - \mathbf{x}_n = \mathbf{e}_{n+1} - \mathbf{e}_n = -\mathbf{e}_n + O(\|\mathbf{e}_n\|^2).$$

So, as the iteration converges, the norm of the step is asymptotically equal to the norm of the previous error. Suppose, for example, one wishes to terminate the iteration when $\|\mathbf{e}_n\| \le \tau$. One could very safely stop the iteration when $\|\mathbf{x}_{n+1} - \mathbf{x}_n\| \le \tau$ and return $\mathbf{x}_{n+1}$ as the solution. Alternatively, one could terminate the iteration when $\|\mathbf{x}_n - \mathbf{x}_{n-1}\| \le \alpha\sqrt{\tau}$, where $\alpha$ is a small constant. Then quadratic convergence would imply that

$$\|\mathbf{e}_n\| = O(\alpha^2 \tau),$$

which would suffice if $\alpha$ were small enough to balance the constant in the $O$-term.

If the iteration converges q-superlinearly, then

$$\|\mathbf{s}\| = \|\mathbf{x}_+ - \mathbf{x}_c\| = \|\mathbf{e}_c\| + o(\|\mathbf{e}_c\|)$$

and the step is still an excellent surrogate for the error in the previous iteration, but one has less information than in the quadratically convergent case and cannot use $\|\mathbf{s}\|$ to estimate $\|\mathbf{e}_{n+1}\|$. One can apply similar logic if one has an accurate upper bound for the q-factor in a q-linearly convergent iteration. If one knows that

$$\|\mathbf{e}_+\| \le \alpha \|\mathbf{e}_c\|,$$

then

$$(1-\alpha)\|\mathbf{e}_c\| \le \|\mathbf{s}\|.$$

From this we can recover an estimate for the error in terms of the step,

$$\|\mathbf{e}_+\| \le \alpha \|\mathbf{e}_c\| \le \frac{\alpha}{1-\alpha}\|\mathbf{s}\|.$$

See Petzold (1983) or Ascher and Petzold (1998) for examples of how this can be used in an initial value problem integration, and Tocci, Kelley and Miller (1997) for an example of the limitations of this idea.

The relation of the residual norm to the norm of the error is very similar to that for the linear case. In the linear case the equation is $\mathbf{A}\mathbf{x} = \mathbf{b}$, the residual is $\mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{x}$, and the error is $\mathbf{e} = \mathbf{x} - \mathbf{A}^{-1} b$. The standard result is

$$\kappa(\mathbf{A})^{-1} \frac{\|\mathbf{r}\|}{\|\mathbf{b}\|} \le \frac{\|\mathbf{e}\|}{\|\mathbf{A}^{-1} b\|} \le \kappa(\mathbf{A}) \frac{\|\mathbf{r}\|}{\|\mathbf{b}\|}$$

(Kelley 1995, Demmel 1997, Golub and Van Loan 1996). This familiar estimate compares the residual at $\mathbf{x}$ to the residual at the zero vector. In the nonlinear case it is not generally useful to use the zero vector as a reference, so we will use the initial iterate $\mathbf{x}_0$ instead.

Suppose the standard assumptions hold and $\mathbf{x}_0$ and $\mathbf{x}$ both satisfy (2.3). Then, using the fundamental theorem of calculus, as we did in (2.6),

$$\begin{aligned}\mathbf{F}(\mathbf{x}) &= \int_0^1 \mathbf{F}'(\mathbf{x}^* + t\mathbf{e})\mathbf{e}\,\mathrm{d}t \\ &= \mathbf{F}'(\mathbf{x}^*)\mathbf{e} + \int_0^1 (\mathbf{F}'(\mathbf{x}^* + t\mathbf{e}) - \mathbf{F}'(\mathbf{x}^*))\mathbf{e}\,\mathrm{d}t. \qquad (2.8)\end{aligned}$$

Therefore, using (2.3),

$$\begin{aligned}\|\mathbf{F}(\mathbf{x})\| &\le \|\mathbf{F}'(\mathbf{x}^*)\mathbf{e}\| + \gamma\|\mathbf{e}\|^2/2, \\ &\le \|\mathbf{F}'(\mathbf{x}^*)\mathbf{e}\| + \|\mathbf{F}'(\mathbf{x}^*)^{-1}\|^{-1}\|\mathbf{e}\|/4 \le \frac{5}{4}\|\mathbf{F}'(\mathbf{x}^*)\mathbf{e}\|.\end{aligned}$$

Similarly

$$\|\mathbf{F}(\mathbf{x})\| \ge \frac{3}{4}\|\mathbf{F}(\mathbf{x}^*)\mathbf{e}\|,$$

and the same inequalities hold for $\mathbf{x}_0$. Hence

$$\frac{3}{5}\kappa(\mathbf{F}'(\mathbf{x}^*))^{-1}\frac{\|\mathbf{e}\|}{\|\mathbf{e}_0\|} \le \frac{\|\mathbf{F}(\mathbf{x})\|}{\|\mathbf{F}(\mathbf{x}_0)\|} \le \frac{5}{3}\kappa(\mathbf{F}'(\mathbf{x}^*))\frac{\|\mathbf{e}\|}{\|\mathbf{e}_0\|}. \qquad (2.9)$$

There is nothing magic about the numbers 3/5 and 5/3. Both are artifacts of the fraction 1/2 in (2.3). As $\mathbf{x}$ and $\mathbf{x}_0$ approach $\mathbf{x}^*$, both coefficients will approach 1. So, the inequality (2.9) is satisfyingly consistent with the linear case (where $\gamma = 0$ and $\rho^* = \infty$, so any $\mathbf{x}$ satisfies (2.3)).

Most implementations of Newton's method do not attempt to compute the step with high accuracy, as we will see in the following sections. Instead one accepts low accuracy in the Jacobian, the linear solve for the step, or even the residual itself. In these cases it is usually unwise to terminate on small steps, and one must terminate on small residuals and accept the effects of ill-conditioning. In the descriptions of algorithms, we will terminate when

$$\|\mathbf{F}(\mathbf{x})\| \le \tau_a + \tau_r\|\mathbf{F}(\mathbf{x}_0)\|. \qquad (2.10)$$

### *2.4. Implementation: LU factorization of* $\mathbf{F}'$

The outline of a Newton iteration is simple. One evaluates the residual, computes the step, and continues until a termination criterion is satisfied. A broad outline of the Newton iteration is shown in algorithm **newton**.

```
newton($\mathbf{x}, \mathbf{F}$)
    Evaluate $\mathbf{F}(\mathbf{x})$; terminate?
    Solve $\mathbf{F}'(\mathbf{x})\mathbf{s} = -\mathbf{F}(\mathbf{x})$
    $\mathbf{x} \leftarrow \mathbf{x} + \mathbf{s}$
```

Algorithm **newton** leaves out all the important details. We will generally terminate the Newton iteration when the residual norm is small using (2.10). However, that alone is not enough. One must limit the number of iterations to avoid an infinite loop when, for example, the equation has no solution. One must also decide how to solve the linear equation for the step. If the Jacobian $\mathbf{F}'$ is small, dense and unstructured, the natural implementation of algorithm **newton** is to use Gaussian elimination and compute an LU factorization of $\mathbf{F}'$. The resulting algorithm, **newton_LU**, is now quite specific and the reader should be able to implement it easily.

```
newton_LU($\mathbf{x}, \mathbf{F}, \tau_a, \tau_r, maxit$)
    $itc = 0$
    evaluate $\mathbf{F}(\mathbf{x})$; $\tau \leftarrow \tau_r \|\mathbf{F}(\mathbf{x})\| + \tau_a$.
    while $\|\mathbf{F}(\mathbf{x})\| > \tau$ and $itc < maxit$ do
        compute $\mathbf{F}'(\mathbf{x})$; factor $\mathbf{F}'(\mathbf{x}) = \mathbf{LU}$
        solve $\mathbf{LUs} = -\mathbf{F}(\mathbf{x})$
        $\mathbf{x} \leftarrow \mathbf{x} + \mathbf{s}$
        evaluate $\mathbf{F}(\mathbf{x})$
        $itc \leftarrow itc + 1$
    end while
```

Algorithm **newton_LU** works well and is widely used. However, there is more to consider. One important matter is how one computes $\mathbf{F}'$. The best way, if one can do it, is to compute the Jacobian analytically. Analytic Jacobians are usually less expensive computationally and avoid any possible problems with differencing. Computing analytic Jacobians is not possible in a general-purpose code, however, and a typical approach is to compute $\mathbf{F}'$ with a forward difference. One way to do this is to approximate the $j$th column of $\mathbf{F}'(\mathbf{x})$ with the difference

$$\frac{\mathbf{F}(\mathbf{x} + h\mathbf{u}_j) - \mathbf{F}(\mathbf{x})}{h}, \tag{2.11}$$

where $\mathbf{u}_j$ is the unit vector in the $j$th coordinate direction. A finite-difference Jacobian $\nabla_h \mathbf{F}(\mathbf{x})$, therefore, has an $O(h)$ error. The computational cost is $N$ additional function evaluations, one for each direction. If the cost of an evaluation of $\mathbf{F}$ is $O(N^2)$, as it would be for a linear equation, then the cost of computing the finite-difference Jacobian is $O(N^3)$, which is the same order as LU factorization. Hence, the construction and factorization of the Jacobian can dominate the cost of the solve. We will show later that analytic Jacobians and finite-difference Jacobians usually produce very similar Newton iterations. The difference is in the expense of computing the finite-difference Jacobian. If the Jacobian is sparse, there are ways to do the differencing much more efficiently. Curtis, Powell and Reid (1974) describe one such method.

The *chord method* modifies the Newton iteration by moving the Jacobian evaluation and factorization out of the loop. The coefficient matrix in the linear equation for the step is $\mathbf{F}(\mathbf{x}_0)$ for every nonlinear iteration. For example, algorithm **newton_LU** becomes an implementation of the chord method by moving one line, shown as algorithm **chord_LU**.

```
chord_LU($\mathbf{x}, \mathbf{F}, \tau_a, \tau_r, maxit$)
  $itc = 0$
  evaluate $\mathbf{F}(\mathbf{x})$; $\tau \leftarrow \tau_r \|\mathbf{F}(\mathbf{x})\| + \tau_a$.
  compute $\mathbf{F}'(\mathbf{x})$; factor $\mathbf{F}'(\mathbf{x}) = \mathbf{LU}$
  while $\|\mathbf{F}(\mathbf{x})\| > \tau$ and $itc < maxit$ do
    solve $\mathbf{LUs} = -\mathbf{F}(\mathbf{x})$
    $\mathbf{x} \leftarrow \mathbf{x} + \mathbf{s}$
    evaluate $\mathbf{F}(\mathbf{x})$
    $itc \leftarrow itc + 1$
  end while
```

The chord iteration is

$$\mathbf{x}_+ = \mathbf{x}_c - \mathbf{F}'(\mathbf{x}_0)^{-1}\mathbf{F}(\mathbf{x}_c),$$

so one has simply replaced the Jacobian at the current point with an approximation. The error in that approximation is

$$\|\mathbf{F}'(\mathbf{x}_c) - \mathbf{F}'(\mathbf{x}_0)\| \le \gamma \|\mathbf{x}_c - \mathbf{x}_0\| = \gamma \|\mathbf{e}_c - \mathbf{e}_0\|.$$

One can prove local convergence with the standard assumptions if the initial iterate is sufficiently near $\mathbf{x}^*$. However, (2.3) may not be good enough. The next section looks at a longer list of approximations one can apply to Newton's method and their effects on the iteration.

### 2.5. *Errors in* $\mathbf{F}$ *and* $\mathbf{F}'$

Suppose one approximates Newton's method by

$$\mathbf{x}_+ = \mathbf{x}_c + \mathbf{s},$$

where

$$\|\mathbf{J}_c \mathbf{s} + (\mathbf{F}(x_c) + \epsilon(\mathbf{x}_c))\| \le \eta_c \|\mathbf{F}(x_c) + \epsilon(\mathbf{x}_c)\| \tag{2.12}$$

and

$$\|\mathbf{J}_c - \mathbf{F}'(\mathbf{x}_c)\| \le \Delta_c. \tag{2.13}$$

We allow for errors in every possible way in this approximation. The evaluation of $\mathbf{F}$ has an error $\epsilon$. We have an approximate Jacobian $\mathbf{J}$ for the linear equation for the Newton step. Finally, we do not even solve that incorrect equation for the step exactly, rather we take a step $\mathbf{s}$ which satisfies the *inexact Newton condition*

$$\|\mathbf{J}_c\mathbf{s} + \mathbf{F}(\mathbf{x}_c) + \epsilon(\mathbf{x}_c)\| \leq \eta_c\|\mathbf{F}(\mathbf{x}_c) + \epsilon(\mathbf{x}_c)\| \tag{2.14}$$

(Dembo, Eisenstat and Steihaug 1982). One way to interpret the inexact Newton condition is as the termination criterion for an iterative linear solver (small relative residuals).

One should expect the iteration to converge if the standard assumptions hold and the errors are sufficiently small. Theorem 2.3 (Kelley 1995) quantifies that.

**Theorem 2.3.** Let Assumption 2.1, (2.3), and (2.14) hold. Then

$$\|\mathbf{e}_+\| = O(\|\mathbf{e}_c\|^2 + (\|\eta_c\| + \Delta_c)\|\mathbf{e}_c\| + \|\epsilon(\mathbf{x}_c)\|). \tag{2.15}$$

Theorem 2.3 is very satisfying and explains most of the algorithms used in practice. We will now apply (2.15) to several examples.

For the chord method, $\epsilon = 0$, $\eta = 0$ and $\mathbf{J} = \mathbf{F}'(\mathbf{x}_0)$. Hence

$$\Delta_c = \|\mathbf{F}'(\mathbf{x}_0) - \mathbf{F}'(\mathbf{x}_c)\| \leq \gamma\|\mathbf{x}_0 - \mathbf{x}_c\| \leq \gamma(\|\mathbf{e}_0\| + \|\mathbf{e}_c\|).$$

If (2.3) holds, then $\|\mathbf{e}_1\| \leq \|\mathbf{e}_0\|/2$, because the first chord iteration is a Newton iteration. However, one needs a better initial iterate to compensate for the error in the Jacobian. If the initial iterate is sufficiently good, then

$$\|\mathbf{e}_{n+1}\| = O(\|\mathbf{e}_n\|^2 + \|\mathbf{e}_0\|\|\mathbf{e}_n\|) = O(\|\mathbf{e}_0\|\|\mathbf{e}_n\|) < \|\mathbf{e}_n\|.$$

Hence, the convergence of the chord method is not q-quadratic, but rather q-linear, with a q-factor proportional to $\|\mathbf{e}_0\|$.

Theorem 2.3 is also the tool one needs to understand the effects of approximating the Jacobian with finite differences. For this we will assume that $\epsilon$ is independent of $\mathbf{x}$, $\eta = 0$, and the error in the Jacobian is first-order in the difference increment. The statement

$$\|\nabla_h\mathbf{F}(\mathbf{x}) - \mathbf{F}'(\mathbf{x})\| = O(h)$$

hides the prefactor of $\gamma/2$ in the $O$-term. If $\gamma$, the Lipschitz constant of $\mathbf{F}'$, is not too large, then a finite-difference Jacobian can be used safely. Most of the time this approximation is fine, but there are exceptions (Kerkhoven and Jerome 1990, Coughran and Jerome 1990).

Theorem 2.3 tells us that in this case

$$\|\mathbf{e}_+\| = O(\|\mathbf{e}_c\|^2 + h\|\mathbf{e}_c\| + \|\epsilon\|).$$

The estimate implies that we cannot hope to reduce $\|\mathbf{e}\|$ to any less than $\|\epsilon\|$ and that the error terms balance when $h = O(\sqrt{\|\epsilon\|})$ (which is a standard lesson in numerical analysis about finite-difference derivatives: Kelley 1995). The more subtle message in the estimate is that if $h = O(\sqrt{\|\epsilon\|})$, the iteration is indistinguishable from the Newton iteration with an exact derivative until $\|\mathbf{e}\| \approx \sqrt{\|\epsilon\|}$.

As a final example, we will consider the secant method for scalar equations (*i.e.* equations for one variable) $f(x) = 0$. Here the model derivative is

$$j_c = \frac{f(x_c) - f(x_-)}{x_c - x_-},$$

where $x_-$ is the iterate before $x_c$. One must, of course, decide what $x_{-1}$ should be. One good choice is $1.01 \times x_0$, which we will use in the examples. Similarly to the analysis of the chord method, we have

$$|j_c - f'(x_c)| = O(|e_c| + |e_-|).$$

Theorem 2.3 with $\epsilon = 0$ and $\eta = 0$ says that if $|e_c| \leq |e_-|$ are sufficiently small,

$$|e_+| = O(|e_c||e_-|). \tag{2.16}$$

So, if the initial iterations are sufficiently good, the secant iteration converges and $|e_{n+1}| = O(|e_n||e_{n-1}|)$, which implies that the convergence is *q-superlinear*, that is,

$$\lim_{n\to\infty} \frac{|e_{n+1}|}{|e_n|} = 0. \tag{2.17}$$

The secant method is limited to scalar equations. In fact, the secant method converges q-superlinearly with q-order $\alpha = (1+\sqrt{5})/2$, that is,

$$|e_+| = O(|e_c|^\alpha).$$

The quasi-Newton methods, which we discuss in Section 2.8, extend the secant method to systems of equations. Scalar equations are no longer an active topic of research, but at one time there was considerable activity (Traub 1964). We will have very little to say about them in this article aside from a few examples.

We can illustrate these results with a simple example. The residual for the scalar equation

$$f(x) = x - \mathrm{e}^{-x}\cos(x) = 0$$

can be evaluated to about 15 figures of accuracy. The Newton iteration is

$$x_+ = x_c - f(x_c)/(1 + \mathrm{e}^{-x_c}(\sin(x_c) + \cos(x_c))).$$

Using the initial iterate $x_0 = 1$, we compare the iteration histories of Newton's method, Newton's method with a forward difference derivative, and the chord method. We will use both a table and a plot, and invite the reader to pick her or his own favourite way to present such data.

To illustrate the effects of the error in $f$, we tabulate and plot $|f|$ for the three methods. We set the termination criteria to $\tau_a = \tau_r = 10^{-20}$. Since

Table 2.1. Iteration histories: $|f(x_n)|$.

| $n$ | Newton | FD Newton | Chord | Secant |
|---|---|---|---|---|
| 0 | $8.0123\times10^{-1}$ | $8.0123\times10^{-1}$ | $8.0123\times10^{-1}$ | $8.0123\times10^{-1}$ |
| 1 | $8.9455\times10^{-2}$ | $8.9455\times10^{-2}$ | $8.9455\times10^{-2}$ | $9.1464\times10^{-2}$ |
| 2 | $6.7756\times10^{-4}$ | $6.7756\times10^{-4}$ | $1.8716\times10^{-2}$ | $8.1187\times10^{-3}$ |
| 3 | $4.1187\times10^{-8}$ | $4.1175\times10^{-8}$ | $3.7460\times10^{-3}$ | $6.4885\times10^{-5}$ |
| 4 | $1.1102\times10^{-16}$ | $5.5511\times10^{-16}$ | $7.5704\times10^{-4}$ | $4.7404\times10^{-8}$ |
| 5 | $2.2204\times10^{-16}$ | $1.1102\times10^{-16}$ | $1.5270\times10^{-4}$ | $2.7611\times10^{-13}$ |
| 6 | $1.1102\times10^{-16}$ | $2.2204\times10^{-16}$ | $3.0813\times10^{-5}$ | $1.1102\times10^{-16}$ |
| 7 | $2.2204\times10^{-16}$ | $1.1102\times10^{-16}$ | $6.2172\times10^{-6}$ | $2.2204\times10^{-16}$ |
| 8 | $1.1102\times10^{-16}$ | $2.2204\times10^{-16}$ | $1.2545\times10^{-6}$ | $1.1102\times10^{-16}$ |
| 9 | $2.2204\times10^{-16}$ | $1.1102\times10^{-16}$ | $2.5312\times10^{-7}$ | $1.1102\times10^{-16}$ |
| 10 | $1.1102\times10^{-16}$ | $2.2204\times10^{-16}$ | $5.1072\times10^{-8}$ | NaN |

$\epsilon \approx 10^{-15}$ in this example, the theory does not imply that it is possible to drive the residual to a value as small as $10^{-20}$, and the computation confirms that. We used a difference increment of $h = 10^{-7}$.

Those of you accustomed to looking at columns of figures may have noticed that, as the theory predicts, there is very little difference between the finite-difference Newton method and the version with analytic derivatives until the iteration *stagnates* at roughly the level of machine precision. One can also notice that the residuals for the chord method decay more slowly, by a factor of 4–5 with each iteration. The secant method converges faster than chord, but not as fast as Newton's method. One weakness of the secant method is exposed by the 10th iteration, where there is a floating-point exception. The problem is that the $f(x_n) = f(x_{n+1})$ and $x_n = x_{n+1}$ at this point, so one gets $j_c = 0/0$, which is reported in IEEE arithmetic (Overton 2001) as NaN (Not a Number). In this example we have run the iteration far beyond any sensible termination point. The NaN and the stagnation in the Newton iteration are signs of that.

It is more illuminating, at least in the author's opinion, to visualize iteration histories, and Table 2.1 is one of the very few tables we will use for that. In Figure 2.1 we visualize the data from the table in a semi-log plot. It is very clear that there is little difference between the two realizations of Newton's method in terms of the number of nonlinear iterations needed to converge to the limiting level of precision. One can also see the signature of superlinear convergence in downward concavity of the residual history for Newton's method and the secant method. The NaN is missing from the plot by convention. The q-linear convergence of the chord method appears as a linear residual history when plotted in this way.

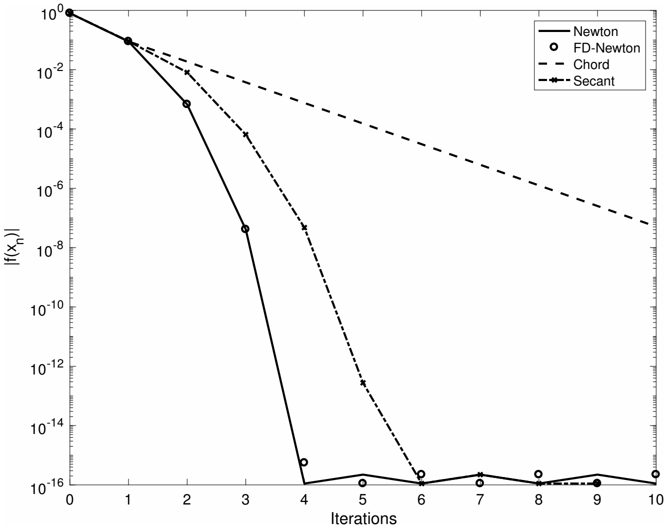

Figure 2.1. Visual iteration histories.

As a general rule, if residuals are accurate to machine unit roundoff, the iterations will stagnate at that level. There are exceptions. For example, if the floating-point implementation of a function has a root in the floating-point number system, then one can continue the iteration for much longer. The equation $\arctan(x) = 0$ is an example of this phenomenon.

Our final example in this section is the Shamanskii method (Shamanskii 1967). This method is a hybrid between Newton's method and the chord method. The idea is to update the Jacobian every $m \geq 1$ iterations. Clearly $m = 1$ is Newton's method and $m = \infty$ is the chord method. If Gaussian elimination is the linear solver, the Shamanskii iteration for finite $m$ is **shamanskii_LU**.

In algorithm **shamanskii_LU** the iteration counter is incremented outside of the inner $j$ loop. Keeping this in mind, Theorem 2.3 states that

$$\|\mathbf{e}_{n+1}\| = O(\|\mathbf{e}_n\|^{m+1}). \tag{2.18}$$

If $m > 1$ the convergence rate is faster than the q-quadratic rate for Newton's method. We say the convergence is q-superlinear with *q-order* $m + 1$. The Shamanskii method is less appealing than it appears because if the Jacobian is sufficiently near $\mathbf{F}'(\mathbf{x}^*)$ the modest reduction in the iterations is not worth the cost of computing and factoring the Jacobian. In many cases $m = \infty$ (the chord method) is a better choice (Brent 1973).

This section has looked at the effects of $\Delta$ and $\epsilon$ on the convergence of Newton's method. Typically proofs set $\epsilon = 0$ and proceed as if evaluations $\mathbf{F}$ were exact. The users of the results generally know that $\epsilon \neq 0$ and when the

```
shamanskii_LU($\mathbf{x}, \mathbf{F}, \tau_a, \tau_r, maxit$)
  $itc = 0$
  evaluate $\mathbf{F}(\mathbf{x})$; $\tau \leftarrow \tau_r \|\mathbf{F}(\mathbf{x})\| + \tau_a$.
  while $\|\mathbf{F}(\mathbf{x})\| > \tau$ and $itc < maxit$ do
    compute $\mathbf{F}'(\mathbf{x})$; factor $\mathbf{F}'(\mathbf{x}) = \mathbf{LU}$
    $\mathbf{y} = \mathbf{x}$
    for $j = 1 : m$ do
      solve $\mathbf{LUs} = -\mathbf{F}(\mathbf{y})$
      $\mathbf{x} \leftarrow \mathbf{y} + \mathbf{s}$
      evaluate $\mathbf{F}(\mathbf{y})$
    end for
    $\mathbf{x} = \mathbf{y}$
    $itc \leftarrow itc + 1$
  end while
```

error in the function evaluation is important. We will follow that approach in the next section, where the inexactness in the linear solver, measured by $\eta$, is the important part of the algorithm.

### *2.6. Inexact Newton methods and JFNK*

When one solves a linear equation $\mathbf{Ax} = \mathbf{b}$ with an iterative method, one usually terminates the iteration when the *relative residual* $\|\mathbf{Ax} - \mathbf{b}\|/\|\mathbf{b}\|$ is sufficiently small. An *inexact Newton method* (Dembo *et al.* 1982) approximates Newton's method by using a step $\mathbf{s}$ that satisfies the *inexact Newton condition*

$$\|\mathbf{F}'(\mathbf{x}_c)\mathbf{s} + \mathbf{F}(\mathbf{x}_c)\| \leq \eta_c \|\mathbf{F}(\mathbf{x}_c)\|. \tag{2.19}$$

Here $\eta$ is called the *forcing term.*

While the theory does not say how one realizes (2.19), in practice it is usually the outcome of an iterative method for solving $\mathbf{F}'(\mathbf{x}_c)\mathbf{x} = -\mathbf{F}(\mathbf{x}_c)$. The term *Newton-iterative* method is used in that case. The iteration for $\mathbf{x}$ is called the *outer* or *nonlinear* iteration. The iterative method for the linear equation is called the *inner* or *linear* iteration. A specific linear solver or class of solvers is often specified. For example Newton–Krylov and Newton–GMRES are common choices. Preconditioners can also be part of the name. Newton–Krylov–Schwarz (Cai, Gropp, Keyes and Tidriri 1994) methods use a Krylov linear solver and a Schwarz domain decomposition method as a preconditioner.

A straightforward application of Theorem 2.3 with $\Delta = 0$ and $\epsilon = 0$ leads to

$$\|\mathbf{e}_+\| = O(\|\mathbf{e}_c\|^2 + \eta_c \|\mathbf{e}_c\|). \tag{2.20}$$

The estimate (2.20) leads to a convergence theorem.

**Theorem 2.4.** Let Assumption 2.1 hold. Then if $\mathbf{x}_0$ is sufficiently near $\mathbf{x}^*$, $0 \le \eta_n \le \bar{\eta} < 1$, and $\bar{\eta}$ is sufficiently small, then inexact Newton iteration converges. Moreover, the convergence is

- q-linear,
- q-superlinear if $\eta_n \to 0$, and
- q-quadratic if $\eta_n = O(\|\mathbf{F}(\mathbf{x}_n)\|)$.

There are, as one might suspect from Theorem 2.4, many approaches to managing $\eta$ as the iteration progresses. Similar to the distinction between Newton's method and the chord method, it is rarely a good idea to make $\eta$ very small, especially at the beginning of the iteration when only limited accuracy is needed to get the same reduction in error that one would get with Newton's method itself. While one could make $\eta$ small once the residuals are small, it is not clear that the additional cost in the linear solve makes the reduction in nonlinear iterations worthwhile. There are useful discussions of this issue in Kelley (1995) and Eisenstat and Walker (1996). The author of this article has had success with $\eta_n \equiv 1/10$.

Theorem 2.4 does not specify any particular norm. If one uses the weighed norm

$$\|\mathbf{x}\|_* = \|\mathbf{F}'(\mathbf{x}^*)\mathbf{x}\|,$$

then the theory no longer needs a small $\bar{\eta}$. Any $\bar{\eta} < 1$ will do.

**Theorem 2.5.** Let Assumption 2.1 hold. Then if $\mathbf{x}_0$ is sufficiently near $\mathbf{x}^*$ and $0 \le \eta_n \le \bar{\eta} < \tilde{\eta} < 1$,

$$\|\mathbf{e}_{n+1}\|_* \le \tilde{\eta}\|\mathbf{e}_n\|_*$$

and the other conclusions of Theorem 2.4 hold.

Assuming that $\mathbf{e}_n \neq 0$ for all $n$, Theorem 2.5 implies that

$$\limsup \frac{\|\mathbf{e}_{n+1}\|_*}{\|\mathbf{e}_n\|_*} \le \bar{\eta}.$$

If the linear solver is a Krylov method which only needs Jacobian-vector products, it is not necessary or desirable to compute and store a Jacobian matrix. For example, one can approximate the Jacobian-vector product with a forward difference. Methods of this type are called *Jacobian-free Newton–Krylov* (JFNK) methods. JFNK methods are the most common choice for those large-scale nonlinear equations which come from differential and integral equations. Knoll and Keyes (2004) provide an excellent account of JNFK methods and applications.

The linear solver in a JFNK method often requires preconditioning to work well enough to be useful. In most implementations preconditioning is done at the level of the linear solver. Preconditioning can also be encoded in the nonlinear map itself and it is interesting to examine that. We will let $\mathbf{M}$ be the preconditioner for the linear equation for the Newton step, and assume that $\mathbf{M}$ does not depend on the $\mathbf{x}$, the nonlinear iteration. If we precondition from the left, the equation for the Newton step is transformed into

$$\mathbf{M}\mathbf{F}'(\mathbf{x}_c)\mathbf{s} = -\mathbf{M}\mathbf{F}(\mathbf{x}_c).$$

This is exactly the Newton step for the equation

$$\mathbf{M}\mathbf{F}(\mathbf{x}) = 0.$$

So, in the case of left preconditioning, one can place the preconditioning in the definition of the nonlinear map, replacing $\mathbf{F}$ by $\mathbf{M}\mathbf{F}$, or apply it to the linear equation for the step. In either case, the steps and the iterations will be the same as in the unpreconditioned case. Most of the production codes put the preconditioning in the linear solve and measure the unpreconditioned residual $\mathbf{F}$ when terminating the linear iteration. When one does this, however, the termination criterion for the linear iteration would be

$$\|\mathbf{M}\mathbf{F}'(\mathbf{x}_c)\mathbf{s} + \mathbf{M}\mathbf{F}(\mathbf{x}_c)\| \le \eta\|\mathbf{M}\mathbf{F}(\mathbf{x}_c)\|,$$

which is the inexact Newton condition for $\mathbf{M}\mathbf{F}$. This does not change the theory if $\eta$ is sufficiently small. To summarize, replacing $\mathbf{F}$ with $\mathbf{M}\mathbf{F}$ does not change the iteration or the steps, but does change the norm of the Jacobian at the solution and the Lipschitz constant for the Jacobian.

Similarly, if one preconditions the linear equation from the right, the equation is

$$\mathbf{F}'(\mathbf{x}_c)\mathbf{M}\mathbf{z} = -\mathbf{F}(\mathbf{x}_c), \quad \mathbf{s} = \mathbf{M}\mathbf{z},$$

and the corresponding nonlinear system is

$$\mathbf{F}(\mathbf{M}\mathbf{y}) = 0, \quad \mathbf{x} = \mathbf{M}\mathbf{y}.$$

In this case the residuals are unchanged and the inexact Newton condition has its original meaning if one puts the preconditioning in the linear solver.

### *2.7. Global convergence*

The Newton iteration for $\arctan(x) = 0$ with $x_0 = 1$ exhibits classic quadratic convergence. With a poor initial iterate, $x_0 = 10$ for example, the first five iterations are

$$10, \quad -138, \quad 2.9 \times 10^4, \quad -1.5 \times 10^9, \quad 9.9 \times 10^{17}.$$

This divergence is consistent with the theory because the initial iterate is so poor. The *Armijo line search* (Armijo 1966) is a wonderful and simple solution to this problem.

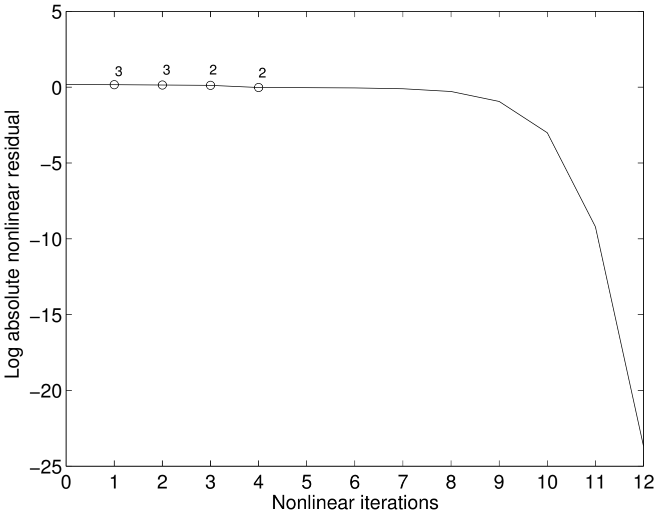

Figure 2.2. The Armijo line search for $\arctan(x) = 0$, $x_0 = 10$.

The idea is to observe that even though the Newton step moves farther from the root, the direction is correct. To clarify this we make a distinction between the Newton direction

$$\mathbf{d} = -\mathbf{F}'(\mathbf{x}_c)^{-1}\mathbf{F}(\mathbf{x}_c)$$

and the Newton step

$$\mathbf{s} = \mathbf{x}_+ - \mathbf{x}_c.$$

The simplest strategy is to find the least $\lambda = 2^{-m}$ for $m = 0, 1, \dots$ so that

$$\|\mathbf{F}(\mathbf{x}_c + \lambda\mathbf{d})\| < \|\mathbf{F}(\mathbf{x}_c)\| \tag{2.21}$$

and use $\mathbf{s} = \lambda\mathbf{d}$. The *simple decrease* condition (2.21) is a bit too weak for a convergence analysis, but is close enough to save the arctan iteration, as one can see in Figure 2.2.

In the figure the circles are iterations for which the step length required reduction, and the number of stepsize reductions are indicated next to the circles. After the fourth iteration the iteration took full steps, and the local theory holds. Note that the residual reduction continued past the resolution of the floating-point system. The reason for this, as we mentioned in Section 2.5, is that $x^* = 0$ is also a solution of the equation as implemented in MATLAB with IEEE floating-point arithmetic.

Convergence theory requires more than simple decrease as in (2.21). The *sufficient decrease* condition that one needs for theory is

$$\|\mathbf{F}(\mathbf{x}_c + 2^{-m}\mathbf{d})\| < (1 - \alpha 2^{-m})\|\mathbf{F}(\mathbf{x}_c)\|. \tag{2.22}$$

In most of the literature and codes, $\alpha = 10^{-4}$. In most cases sufficient decrease conditions such as (2.22) will lead to the same stepsize decisions as simple decrease, and the purpose is to enable theory. It is common to adaptively change the stepsize reduction factor. One way to do this (Kelley 1995, Dennis and Schnabel 1996) is to begin with a stepsize of $\lambda = 1$, and with each failure of the sufficient decrease condition

$$\|\mathbf{F}(\mathbf{x}_c + \lambda \mathbf{d})\| < (1 - \alpha\lambda)\|\mathbf{F}(\mathbf{x}_c)\|, \tag{2.23}$$

reduce $\lambda$ by a factor $\sigma \in [0.1, 0.5]$. The standard way to do this is to use this history of failed steps to generate a polynomial approximation of $\phi(\lambda) = \|\mathbf{F}(\mathbf{x}_c + \lambda \mathbf{d})\|$ and minimize that polynomial. This adaptivity is very useful in practice.

Algorithm **newton_armijo** is an inexact formulation of the Newton–Armijo method. It includes the essential ideas and makes the theory easy to state. It is critical that one uses $\mathbf{F}'(\mathbf{x})$ and not an approximation. The Armijo rule is not, for example, theoretically supported for the chord method. Adding a line search to a Newton code is easy and requires only a few new lines.

```
newton_armijo($\mathbf{x}, \mathbf{F}, \tau_a, \tau_r$)
  evaluate $\mathbf{F}(\mathbf{x})$; $\tau \leftarrow \tau_r|\mathbf{F}(\mathbf{x})| + \tau_a$.
  while $\|\mathbf{F}(\mathbf{x})\| > \tau$ do
    Find $d$ such that $\|\mathbf{F}'(\mathbf{x})\mathbf{d} + \mathbf{F}(\mathbf{x})\| \le \eta\|\mathbf{F}(\mathbf{x})\|$
    If no such $d$ can be found, terminate with failure.
    $\lambda = 1$
    while $\|\mathbf{F}(\mathbf{x} + \lambda\mathbf{d})\| > (1 - \alpha\lambda)\|\mathbf{F}(\mathbf{x})\|$ do
      $\lambda \leftarrow \sigma\lambda$ where $\sigma \in [1/10, 1/2]$ is computed by minimizing a poly-
      nomial model of $\|\mathbf{F}(\mathbf{x} + \lambda\mathbf{d})\|^2$.
    end while
    $\mathbf{x} \leftarrow \mathbf{x} + \lambda\mathbf{d}$
  end while
```

Algorithm **newton_armijo** does not say how or if the forcing term $\eta$ changes with the iteration. Most of the codes do vary $\eta$ and use the ideas in Eisenstat and Walker (1996). One example from Eisenstat and Walker (1996) which is common is

$$\eta_n = \begin{cases} \eta_{\max} & n = 0, \\ \min(\eta_{\max}, \eta_n^A) & n > 0. \end{cases} \tag{2.24}$$

Here

$$\eta_n^A = \gamma_\eta \|F(x_n)\|^2 / \|F(x_{n-1})\|^2$$

and $\gamma_\eta$ is a parameter. The parameter $\eta_{\max}$ is an upper limit on the sequence $\{\eta_n\}$. Eisenstat and Walker (1996) use the choices $\gamma_\eta = 0.9$ and $\eta_{\max} = 0.9999$. The author of this article, however, likes the choice $\eta_n \equiv 0.1$.

Theorem 2.6 is very satisfying. For sufficiently smooth problems, the Newton–Armijo iteration has only three possible outcomes. One is convergence to a solution which satisfies the standard assumptions. In that case the stepsize $\lambda$ will be one in the terminal phase of the iteration. The other two outcomes are failures, which are easy to detect numerically. One failure mode is that the iteration becomes unbounded. An example of such a problem is the scalar equation $e^x = 0$, where the Newton–Armijo iteration diverges to $-\infty$. The second failure mode is that the Jacobian drifts to singularity. An example is the scalar equation $x^2 + 1 = 0$.

**Theorem 2.6.** Suppose $\mathbf{F}$ is Lipschitz continuously differentiable, $\{\mathbf{x}_n\}$ is the inexact Newton–Armijo sequence, and $0 < \eta_n < \bar{\eta} < 1$. Then there are only three possibilities.

- $\{\mathbf{x}_n\}$ converges to a root $\mathbf{x}^*$ of $\mathbf{F}$ at which the standard assumptions hold, full steps ($\lambda = 1$) are taken for $n$ sufficiently large, and the local convergence theory holds.
- The sequence $\{\mathbf{x}_n\}$ is unbounded.
- The sequence $\{\mathbf{F}'(\mathbf{x}_n)^{-1}\}$ is unbounded.

The Newton–Armijo method does not solve all problems. Even in the successful case, there is no guarantee that the iteration converges to a useful solution. Nonlinear equations can have multiple solutions, and there are often constraints such as dynamic stability or correct signs for physical quantities to which the Newton iteration is oblivious. We will consider a few ways to address non-uniqueness in Sections 4 and 7.

Another approach to globalization is that of *trust region* methods (Powell 1970). These methods are widely used in optimization (Conn, Gould and Toint 2000), but less so for nonlinear equations. The idea is to model $\|\mathbf{F}(\mathbf{x}_c)\|_2^2$ with a quadratic and minimize that quadratic in a bounded set, the trust region, centred at $x_c$. We will not discuss these methods in detail here. Absil, Baker and Gallivan (2007) and Higham (1999) report on some interesting applications which are connected to the continuation methods in Section 7.

### *2.8. Broyden's method*

Quasi-Newton methods construct a model Jacobian from the history of the iteration. One maintains both an approximation $\mathbf{x}_n$ of the solution and an approximation $\mathbf{B}_n$ of the Jacobian. There are many of these methods (Kelley 1995, Dennis and Schnabel 1996, Dennis and Walker 1981) and they are widely used in optimization. They have largely been replaced by JFNK methods for nonlinear equations. We will briefly discuss Broyden's method, the simplest of them and the one that is used in the NOX code from Trilinos (Heroux *et al.* 2005).

The Broyden update is

$$\mathbf{B}_+ = \mathbf{B}_c + \frac{(\mathbf{y} - \mathbf{B}_c\mathbf{s})\mathbf{s}^T}{\mathbf{s}^T\mathbf{s}}. \tag{2.25}$$

Here $\mathbf{y} = \mathbf{F}(\mathbf{x}_+) - \mathbf{F}(\mathbf{x}_c)$ and $\mathbf{s} = \mathbf{x}_+ - \mathbf{x}_c$. One can think of this as a generalization of the secant method for scalar equations. For the secant equation, the model derivative at the new point $x_+$ is

$$b_+ = \frac{f(x_+) - f(x_c)}{x_+ - x_c} = \frac{y}{s},$$

giving $b_+ s = y$. For systems of equations, the secant equation

$$\mathbf{B}_+\mathbf{s} = \mathbf{y}$$

is a system of $N$ equations in $N^2$ unknowns. This enables construction of updates that satisfy structural constraints such as sparsity or positivity. Dennis and Schnabel (1996) and Kelley (1995) discuss several kinds of secant updates for nonlinear equations. In optimization, for example, one often wants the model Hessian to be symmetric and positive definite. There are many quasi-Newton update which do that (Dennis and Schnabel 1996, Kelley 1999, Nocedal and Wright 1999). One can also design updates to capture functional analytic properties of infinite-dimensional problems and their discretizations (Kelley and Sachs 1993, Kelley and Sachs 1995, Kelley and Sachs 1987, Kelley and Sachs 1989, Hart and Soul 1973). The most general accounts of theory can be found in Dennis and Walker (1981) and Dennis and Schnabel (1979).

One can implement the Broyden update by storing two vectors for each iteration and using the Sherman–Morrison formula (Kelley 1995) to update the product of $\mathbf{B}^{-1}$ and a vector. The storage burden can be reduced to one vector per nonlinear iteration by using the dependence of $\mathbf{y}$ on $\mathbf{s}$ (Kelley 1995, Deuflhard, Freund and Walter 1990). JFNK methods place the storage burden on the linear iteration, and that seems to be best.

The formula for the update (2.25) allows for a line search in which

$$\mathbf{s} = -\lambda\mathbf{B}_c^{-1}\mathbf{F}(\mathbf{x}_c).$$

Theorem 2.6 does not apply to a Broyden–Armijo algorithm patterned after algorithm **newton_armijo**, but in practice such an algorithm often works fine.

The convergence theory is only local and requires the standard assumptions and accurate initial approximations to the solution and the Jacobian at the solution.

**Theorem 2.7.** If the standard assumptions hold and $\mathbf{x}_0$ and $\mathbf{B}_0$ are sufficiently near $\mathbf{x}^*$ and $\mathbf{F}'(\mathbf{x}^*)$, then $\mathbf{x}_n \to \mathbf{x}^*$ q-superlinearly:

$$\lim_{n\to\infty} \frac{\|\mathbf{e}_{n+1}\|}{\|\mathbf{e}_n\|} = 0.$$

Broyden's method and JFNK methods have remarkable similarities when applied to discretizations of infinite-dimensional problems. The preconditioning issues are very closely related; see, for example, Kelley and Sachs (1985), Kelley and Xue (1996), Burmeister (1975) and Nevanlinna (1993).

### *2.9. Fréchet and Gâteaux derivatives*

We have not fully explained how the results in this section map to the infinite-dimensional setting. The reason for this is that we have defined the derivative in the context of its matrix representation as a Jacobian matrix. We are now at a point where we must make a coordinate-free definition to consider our first infinite-dimensional example in the next section.

If $D \subset \mathbb{R}^M$, we will let $D^\circ$ denote the interior of $D$.

**Definition 2.8.** A function $\mathbf{F} : D \subset \mathbb{R}^N \to \mathbb{R}^M$ is Fréchet differentiable at $x \in D^\circ$ if there is a linear map $\mathbf{F}'(\mathbf{x})$ from $\mathbb{R}^M$ to $\mathbb{R}^N$ such that

$$\lim_{\mathbf{h}\to 0} \frac{\|\mathbf{F}(\mathbf{x}+\mathbf{h}) - \mathbf{F}(\mathbf{x}) - \mathbf{F}'(\mathbf{x})\mathbf{h}\|}{\|\mathbf{h}\|} = 0. \tag{2.26}$$

$\mathbf{F}'$ is called the Fréchet derivative (or simply the derivative) of $\mathbf{F}$ at $\mathbf{x}$.

Fréchet differentiability simply means that the difference quotients converge to the derivative uniformly in the direction $\mathbf{h}/\|\mathbf{h}\|$ as $\|\mathbf{h}\| \to 0$. The Jacobian matrix is simply the matrix representation of the Fréchet derivative in the basis of coordinate directions. So the $j$th column of the Jacobian is $\mathbf{F}'(\mathbf{x})\mathbf{u}_j$ as expressed in (2.11). The results on convergence of Newton's method in this section do not change in the infinite-dimensional case. For quasi-Newton methods, such as Broyden's method, superlinear convergence depends on compactness properties (Kelley and Sachs 1985) of $\mathbf{I} - \mathcal{F}'$, which are trivially satisfied in finite-dimensional problems.

One useful way to compute the Fréchet derivative is to apply (2.26) to compute $\mathbf{F}'(\mathbf{x})\mathbf{u}$ for an arbitrary $\mathbf{u}$. Often one can easily extract $\mathbf{F}'$ directly by looking at the results, since

$$\mathbf{F}'(\mathbf{x})\mathbf{u} = \frac{\mathrm{d}}{\mathrm{d}t}\mathbf{F}(\mathbf{x}+t\mathbf{u})\bigg|_{t=0}. \tag{2.27}$$

As an example, suppose that

$$\mathbf{F}(\mathbf{x}) = \mathbf{A}\mathbf{x} + f(\mathbf{x}),$$

where $\mathbf{A}$ is a linear operator and $f(\mathbf{x})$ is the substitution operator

$$f(\mathbf{x}) \equiv \begin{pmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_N) \end{pmatrix}.$$

Then (2.27) implies that

$$\mathbf{F}'(\mathbf{x}) = \mathbf{A} + \mathrm{diag}(f'(\mathbf{x})).$$

If it exists, the limit

$$\mathrm{d}\mathbf{F}(\mathbf{x} : \mathbf{u}) = \lim_{t \downarrow 0} \frac{\mathbf{F}(\mathbf{x} + t\mathbf{u}) - \mathbf{F}(\mathbf{x})}{t} \tag{2.28}$$

is called the *directional derivative* of $\mathbf{F}$ at $\mathbf{x}$ in the direction $\mathbf{u}$. The scalar example $f(x) = |x|$ at $x = 0$ has directional derivatives in all directions, but is not Fréchet differentiable. This example also shows that $\mathrm{d}\mathbf{F}(\mathbf{x} : \mathbf{u})$ need not be linear in $\mathbf{u}$. If the limit

$$\mathrm{d}\mathbf{F}(\mathbf{x} : \mathbf{u}) = \lim_{t \to 0} \frac{\mathbf{F}(\mathbf{x} + t\mathbf{u}) - \mathbf{F}(\mathbf{x})}{t} \tag{2.29}$$

exists, then $\mathrm{d}\mathbf{F}(\mathbf{x} : \mathbf{u})$ is called the Gâteaux derivative of $\mathbf{F}$ at $\mathbf{x}$ in the direction $\mathbf{u}$. In this case

$$\mathrm{d}\mathbf{F}(\mathbf{x} : \mathbf{u}) = \frac{\mathrm{d}}{\mathrm{d}t} \mathbf{F}(\mathbf{x} + t\mathbf{u}) \bigg|_{t=0},$$

but still need not be linear in $\mathbf{u}$. With the exception of Sections 6 and 7.1, all functions in this article will be Fréchet differentiable. If $\mathbf{F}$ is Fréchet differentiable at $\mathbf{x}$, then $\mathrm{d}\mathbf{F}(\mathbf{x} : \mathbf{u}) = \mathbf{F}'(\mathbf{x})\mathbf{u}$ for all $\mathbf{u}$.

### *2.10. Example: Chandrasekhar H-equation*

This is the first example of an operator in a function space. We will denote such operators by script upper-case letters to distinguish them from their discretizations. The discretizations will be functions on $\mathbb{R}^N$ and will have the boldface fonts we have been using.

The Chandrasekhar $H$-equation (Chandrasekhar 1960, Busbridge 1960) is

$$\mathcal{F}(H)(\mu) = H(\mu) - \left(1 - \frac{\omega}{2} \int_0^1 \frac{\mu H(\nu)\, \mathrm{d}\nu}{\mu + \nu}\right)^{-1} = 0. \tag{2.30}$$

The equation arises in radiative transfer theory and is a very tractable example of a nonlinear equation in a function space. We will regard $\mathcal{F}$ as a map on $C[0,1]$, the Banach space of continuous function on the interval $[0,1]$ with the $\|\cdot\|_\infty$-norm. The unknown is a function $H \in C[0,1]$; $\omega \in [0,1]$ is a parameter.

We will discretize the equation with the composite midpoint rule

$$\int_0^1 f(\mu)\,\mathrm{d}\mu \approx \frac{1}{N}\sum_{j=1}^{N} f(\mu_j),$$

where $\mu_i = (i-1/2)/N$ for $1 \leq i \leq N$.

This leads to an equation in $\mathbb{R}^N$:

$$\mathbf{F}^N(\mathbf{h})_i = h_i - \left(1 - \frac{\omega}{2N}\sum_{j=1}^{N}\frac{\mu_i h_j}{\mu_i+\mu_j}\right)^{-1}. \tag{2.31}$$

We will discuss some important properties of this equation in the context of the infinite-dimensional problem. Our conclusions will be equally valid for the discrete problem because the midpoint rule integrates constant functions exactly.

### *2.10.1. The continuous problem*

Our first task is to differentiate $\mathcal{F}$. Applying (2.26), we see that for all $v \in C[0,1]$,

$$(\mathcal{F}'(H)w) = w - \frac{\omega L w}{(1-\omega L H)^2},$$

where $L$ is the integral operator defined by

$$Lw(\mu) = \frac{1}{2}\int_0^1 \frac{\mu w(\nu)\,\mathrm{d}\nu}{\mu+\nu}.$$

Hence,

$$\mathcal{F}'(H) = I - \frac{\omega L}{(1-\omega L H)^2}.$$

We will now check the standard assumptions. To see that there is a solution, we note that the sequence

$$H_0 = 1, \quad H_{n+1} = 1 + \omega H_n L H_n$$

is increasing. We will show convergence by showing that the $L^1$-norms are bounded:

$$\|H_n\|_1 = \int_0^1 H_n(\nu)\,\mathrm{d}\nu \leq \beta \equiv \frac{1+\sqrt{1-\omega}}{2/\omega}. \tag{2.32}$$

Clearly (2.32) holds for $n=1$. To proceed by induction, suppose $\|H_n\|_1 \leq \beta$. We seek to show that

$$\int_0^1 H_{n+1}(\mu)\,\mathrm{d}\mu = 1 + \frac{\omega}{2}\int_0^1 \frac{\mu H_n(\nu) H_n(\mu)}{\mu+\nu}\,\mathrm{d}\nu \tag{2.33}$$

for all $0 \leq \mu \leq 1$. The trick is to integrate both sides of (2.33) with respect to $\mu$ and note that

$$\int_0^1 \int_0^1 \frac{\mu H_n(\nu) H_n(\mu)}{\mu + \nu} \, d\nu \, d\mu = \int_0^1 \int_0^1 \frac{\nu H_n(\nu) H_n(\mu)}{\mu + \nu} \, d\nu \, d\mu$$
$$= \|H_n\|_1^2 / 2 \leq \beta^2 / 2.$$

We are done since

$$\omega \beta^2 / 4 + 1 \leq \beta$$

by the quadratic formula. So our sequence $H_n$ converges in $L^1$ to a function $H$ which satisfies $H = 1 + \omega H L H$. This implies that $H$ satisfies (2.30), since if $1 - \omega L H$ vanishes at any $\mu \in [0,1]$, then $H(\mu) = 1 + H(\mu)$ implying that $H(\mu) = 0$, which violates the equation since $H(\mu) \geq 1$. Also, since $H_n$ is an increasing sequence, $H$ is a positive function of $\mu$ for $\omega \in [0,1]$.

So $H \in L^1[0,1]$ satisfies (2.30). This implies that $H$ is continuous since $L$ is a bounded operator from $L^1[0,1]$ to $C[0,1]$.

Lipschitz continuity of $\mathcal{F}'$ is easy to check. The most interesting point is non-singularity of $\mathcal{F}'(H)$. If $H$ is the solution of (2.30), then it is clear by the argument above that

$$\int_0^1 H(\mu) \, d\mu = \frac{1 + \sqrt{1 - \omega}}{2/\omega}. \tag{2.34}$$

This clearly shows that there is no real solution for $\omega > 1$.

Since $\mathcal{F}'$ is the sum of a compact integral operator and the identity, singularity of the operator implies that there is a non-trivial null space. Suppose $\mathcal{F}'(H) w = 0$. Then

$$w = \frac{\omega L w}{(1 - \omega L H)^2}.$$

We may take $w \geq 0$ by the Perron–Frobenius theorem (Karlin 1959) and may therefore assume that $\int_0^1 w(\mu) \, d\mu > 0$. Since $H$ is a solution, we have $(1 - \omega L H)^{-1} = H$, and so

$$w(1 - \omega w L H) = \omega H L w.$$

So

$$\int_0^1 w(\mu) \, d\mu = \omega / 2 \int_0^1 w(\mu) \, d\mu \int_0^1 H(\nu) \, d\nu$$

implies that $\omega \|H\|_1 = 2$, which by (2.34) implies that $\omega = 1$.

So $\mathcal{F}'(H)$ is non-singular unless $\omega = 1$. When $\omega = 1$, $\mathcal{F}'(H)$ is indeed singular and, again by the Perron–Frobenius theorem, the null space has dimension one and is spanned by a non-negative function. In fact, that function is $w(\mu) = \mu H(\mu)$. We see that the singularity structure of the $H$-equation is quite simple and will use the $H$-equation as an example again in Section 4.

### *2.10.2. The discrete problem*

Now we return to the discrete version. The purpose of this section is to illustrate how exploiting problem structure can give very different cost estimates from the simple accounting of $O(N^3)$ work for a linear solve and $N$ function evaluations for a Jacobian.

We begin with the cost of an evaluation. Using $\mu_i = (i-1/2)/N$ for $1 \le i \le N$ and (2.31), we obtain

$$\mathbf{F}^N(\mathbf{h})_i = h_i - \left(1 - \frac{\omega}{2N}\sum_{j=1}^{N}\frac{ih_j}{i+j-1}\right)^{-1}. \tag{2.35}$$

The approximate integral operator $\mathbf{L}$, where

$$\mathbf{L}_{ij} = \frac{\omega i}{2N(i+j-1)},$$

is the product of a diagonal and a Hankel matrix (Golub and Van Loan 1996). In fact $\mathbf{L} = \mathbf{D}_1\mathbf{K}$, where

$$\mathbf{D}_1 = \mathrm{diag}(i/2N)$$

and $\mathbf{K}$ is the Hankel matrix

$$\mathbf{K}_{ij} = 1/(i+j-1).$$

So evaluation of $\mathbf{Lh}$ can be done at a cost of $O(N\log N)$ work if one computes the product of the Hankel matrix and a vector with a fast Fourier transform (Golub and Van Loan 1996). Since the remaining cost of the evaluation of $\mathbf{F}^N$ reduces to simple binary operations, the cost of a function evaluation is $O(N\log N)$ work.

We may now proceed exactly as we did in the continuous case. The Jacobian matrix is

$$(\mathbf{F}^N)'(\mathbf{h})_{ij} = \delta_{ij} - \frac{\omega\mathbf{L}_{ij}}{1-\omega(\mathbf{Lu})_i}. \tag{2.36}$$

Hence the Jacobian can be expressed as the identity plus the product of a diagonal and a Hankel matrix. If we set

$$\mathbf{D}_2 = \mathrm{diag}(1-\omega(\mathbf{Lu})_i)^{-1},$$

then

$$(\mathbf{F}^N)'(\mathbf{h}) = \mathbf{I} = \omega\mathbf{D}_1\mathbf{D}_2\mathbf{K},$$

and the Jacobian matrix can be constructed and stored with $O(N^2)$ work and the Jacobian-vector product computed with $O(N\log N)$ work.

One conclusion that may be unexpected is that, even if one does the matrix–vector product with $\mathbf{K}$ naively for $O(N^2)$ work, the solution with Newton–GMRES is far faster than one using a direct method to factor $\mathbf{F}'$.

Table 2.2. Iteration component costs.

| Function evaluation | Jacobian evaluation | Jacobian-vector product | LU factorization |
|---|---|---|---|
| $O(N\log(N))$ | $O(N^2)$ | $O(N\log(N))$ | $N^3/3+O(N^2)$ |

The integral equation structure can be exploited to show that the number of Krylov iterations per nonlinear iteration can be bounded independently of $N$. Hence, as $N$ becomes large the advantage of Newton–Krylov over Newton–LU grows rapidly. The reader should try this and see. Chandrasekhar (1960) tabulates the solution to several figures.

This problem has a rich structure and we will report the costs of a computation in some detail. Table 2.2 summarizes our discussion on the costs of function evaluations, Jacobian evaluations and Jacobian-vector products as functions of $N$.

We will now solve the $H$-equation for a few values of $\omega$. The problem becomes more difficult as $\omega$ increases. We will use $\omega = 0.5$ (easy), $\omega = 0.99$ (less easy) and $\omega = 1$ (tricky). The initial iterate in all cases will be

$$\mathbf{h}_0 = (1, 1, \dots, 1)^T.$$

We will terminate the iteration when

$$\|\mathbf{F}^N(\mathbf{h}_n)\| \leq 10^{-10}\|\mathbf{F}^N(\mathbf{h}_0)\|,$$

so $\tau_r = 10^{-10}$ and $\tau_a = 0$. We will use a mesh with $N = 1000$ points to begin with and then demonstrate that the performance of all the methods does not vary much as the mesh is refined. We will do this numerically in this chapter and discuss the theory in Section 3.3.

We will first look at Newton's method. The dominant cost for a Newton iteration is the matrix factorization. Therefore Newton's method, even though the number of nonlinear iterations is small, is the most expensive approach for this problem when $\omega < 1$ and the number of mesh points $N$ is large. In Figure 2.3 we plot the relative residual $(\|\mathbf{F}^N(\mathbf{h}_n)\|/\|\mathbf{F}^N(\mathbf{h}_0)\|)$ histories for the three values of $\omega$.

Figure 2.3 reinforces the ideas from this section. The concave plots of residual histories for $\omega < 1$ indicate superlinear convergence. The linear plot for $\omega = 1$, however, shows that the standard assumptions are violated in this case, as we demonstrated above. One can analyse this (Decker and Kelley 1980) and show that the simple scalar equation $x^2 = 0$ fully explains the q-/linear convergence rate. For any $x_0 \neq 0$, the Newton iteration for $x^2 = 0$ is simply

$$x_n = x_{n-1}/2.$$

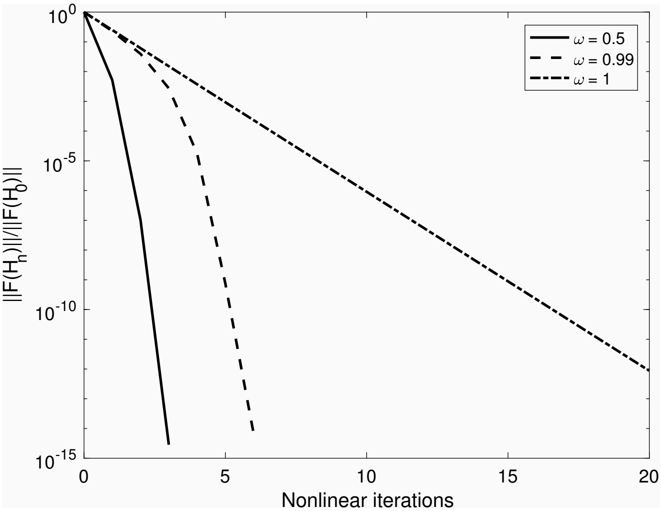

Figure 2.3. Newton's method for the $H$-equation example.

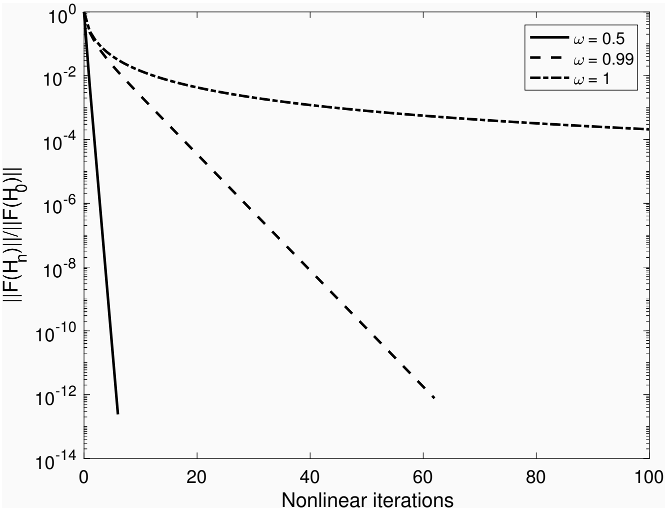

Figure 2.4. The chord method for the $H$-equation example.

Table 2.3. Newton residuals and timings.

| $N$ | $\frac{\Vert\mathbf{F}^N(\mathbf{h}_1)\Vert}{\Vert\mathbf{F}^N(\mathbf{h}_0)\Vert}$ | $\frac{\Vert\mathbf{F}^N(\mathbf{h}_2)\Vert}{\Vert\mathbf{F}^N(\mathbf{h}_0)\Vert}$ | $\frac{\Vert\mathbf{F}^N(\mathbf{h}_3)\Vert}{\Vert\mathbf{F}^N(\mathbf{h}_0)\Vert}$ | Time |
|---|---|---|---|---|
| 1000 | $5.14\times10^{-3}$ | $1.00\times10^{-7}$ | $2.85\times10^{-15}$ | 0.13 |
| 2000 | $5.14\times10^{-3}$ | $1.00\times10^{-7}$ | $2.91\times10^{-15}$ | 0.53 |
| 4000 | $5.14\times10^{-3}$ | $1.00\times10^{-7}$ | $2.68\times10^{-15}$ | 2.47 |
| 8000 | $5.14\times10^{-3}$ | $1.00\times10^{-7}$ | $2.85\times10^{-15}$ | 13.68 |

Hence the convergence is q-linear with q-factor $1/2$. The residuals

$$x_n^2 = x_{n-1}^2/4$$

converge q-linearly to 0 with q-factor $1/4$.

In Figure 2.4 we plot the residual histories for the chord method. For $\omega = 0.5$ and $\omega = 0.99$ one can see the q-linear convergence predicted by theory. However, for $\omega = 1$ something different is happening. The discussion for the continuous case applies without modification to the discrete problem and $\mathbf{F}'(\mathbf{h}^*)$ is singular for $\omega = 1$. There is theory for that case (Decker and Kelley 1983) and the results can be explained by considering the scalar equation $x^2 = 0$. With the initial iterate $x_0 = 1$ the chord iterations are

$$x_n = x_{n-1}(1 - x_{n-1}/2) = O(1/n),$$

which is very slow convergence.

If one refines the mesh for this problem, one will have a more accurate approximation of the solution of the integral equation. However, mesh refinement will have a very small effect on the iteration statistics. Table 2.3 illustrates this by showing some relative residual norms for the Newton iteration with $\omega = 0.5$ and a few values of $N$. As you can see, the relative residual norms are identical to three figures until the terminal iteration. This effect is called *mesh independence*. We will return to this topic in Section 3.3. We were limited to $N = 8000$ because we could not store or factor larger dense matrices in our computing environment.

We also report computer times in seconds using the MATLAB `tic` and `toc` commands. We do this only to show that the run times do not, as you might expect from the $O(N^3)$ work in the matrix factorizations, increase by a factor of eight as we double the size of the problem. The reason for this is that there is a great deal of $O(N^2)$ work in the Jacobian evaluations, which is significant in this case, and the LU factorization in MATLAB is very fast. The effects of the $O(N^2)$ work are also seen in the chord iterations. One would expect the chord iterations to be much faster if the $O(N^3)$ work in the matrix factorization dominated the computation. In Table 2.4 we show the

Table 2.4. Chord residuals and timings.

| $N$ | $\frac{\|\mathbf{F}^N(\mathbf{h}_1)\|}{\|\mathbf{F}^N(\mathbf{h}_0)\|}$ | $\frac{\|\mathbf{F}^N(\mathbf{h}_2)\|}{\|\mathbf{F}^N(\mathbf{h}_0)\|}$ | $\frac{\|\mathbf{F}^N(\mathbf{h}_3)\|}{\|\mathbf{F}^N(\mathbf{h}_0)\|}$ | $\frac{\|\mathbf{F}^N(\mathbf{h}_4)\|}{\|\mathbf{F}^N(\mathbf{h}_0)\|}$ | $\frac{\|\mathbf{F}^N(\mathbf{h}_5)\|}{\|\mathbf{F}^N(\mathbf{h}_0)\|}$ | Time |
|---|---|---|---|---|---|---|
| 1000 | $5.14\times10^{-3}$ | $4.45\times10^{-5}$ | $3.81\times10^{-7}$ | $3.26\times10^{-9}$ | $2.79\times10^{-11}$ | 0.08 |
| 2000 | $5.14\times10^{-3}$ | $4.45\times10^{-5}$ | $3.81\times10^{-7}$ | $3.26\times10^{-9}$ | $2.79\times10^{-11}$ | 0.35 |
| 4000 | $5.14\times10^{-3}$ | $4.45\times10^{-5}$ | $3.81\times10^{-7}$ | $3.26\times10^{-9}$ | $2.79\times10^{-11}$ | 1.51 |
| 8000 | $5.14\times10^{-3}$ | $4.45\times10^{-5}$ | $3.81\times10^{-7}$ | $3.26\times10^{-9}$ | $2.79\times10^{-11}$ | 8.41 |

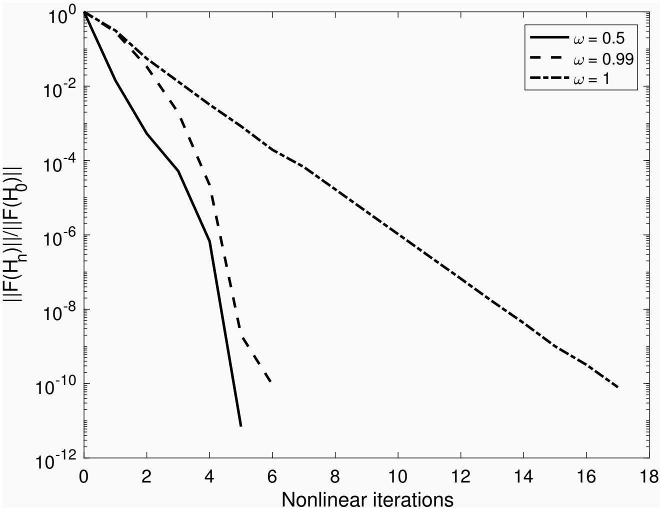

Figure 2.5. Newton–GMRES for the $H$-equation example.

first seven scaled residual norms for the chord iteration and the timings for the entire sequence of 63 iterations. As is the case with Newton's method, the iteration statistics agree. Figures 2.3 and 2.4 would not change with higher values of $N$.

The performance of the JFNK iteration is far better because there is no $O(N^2)$ work or storage at all – only the $O(N \log N)$ cost of the function evaluations. We will illustrate this by repeating the computations above with Newton–GMRES, a finite-difference Jacobian-vector product, and a constant forcing term of $\eta = 0.1$. The results were not affected by switching to an analytic Jacobian-vector product. In Figure 2.5 we plot the iteration histories. The curves are not as smooth as those in Figure 2.3, which reflects the limited accuracy of the linear solves. However, the overall cost was, as you will see in Table 2.5, much less.

Table 2.5. Newton–GMRES residuals and timings.

| $N$ | $\frac{\Vert\mathbf{F}^N(\mathbf{h}_1)\Vert}{\Vert\mathbf{F}^N(\mathbf{h}_0)\Vert}$ | $\frac{\Vert\mathbf{F}^N(\mathbf{h}_2)\Vert}{\Vert\mathbf{F}^N(\mathbf{h}_0)\Vert}$ | $\frac{\Vert\mathbf{F}^N(\mathbf{h}_3)\Vert}{\Vert\mathbf{F}^N(\mathbf{h}_0)\Vert}$ | $\frac{\Vert\mathbf{F}^N(\mathbf{h}_4)\Vert}{\Vert\mathbf{F}^N(\mathbf{h}_0)\Vert}$ | $\frac{\Vert\mathbf{F}^N(\mathbf{h}_5)\Vert}{\Vert\mathbf{F}^N(\mathbf{h}_0)\Vert}$ | Time |
|---|---|---|---|---|---|---|
| 8 000 | $1.43\times10^{-2}$ | $5.28\times10^{-4}$ | $5.22\times10^{-5}$ | $6.70\times10^{-7}$ | $6.95\times10^{-12}$ | 0.02 |
| 16 000 | $1.43\times10^{-2}$ | $5.28\times10^{-4}$ | $5.22\times10^{-5}$ | $6.70\times10^{-7}$ | $6.95\times10^{-12}$ | 0.03 |
| 32 000 | $1.43\times10^{-2}$ | $5.28\times10^{-4}$ | $5.22\times10^{-5}$ | $6.70\times10^{-7}$ | $6.95\times10^{-12}$ | 0.06 |
| 64 000 | $1.43\times10^{-2}$ | $5.28\times10^{-4}$ | $5.22\times10^{-5}$ | $6.70\times10^{-7}$ | $6.95\times10^{-12}$ | 0.09 |

The iteration is so fast that we must solve much larger problems to see the $O(N \log N)$ cost reflected in the timings. Table 2.5 shows the mesh independence of the nonlinear iteration. The performance of the linear iteration is also mesh-independent. For each value of $N$ the entire iteration needed 19 calls to $\mathbf{F}^N$. These 19 calls include both the evaluation of the residual and the additional calls for the finite-difference Jacobian-vector product. One could also use an analytic Jacobian-vector product and both the cost and the results in Table 2.5 would be the same.

## 3. The Kantorovich theorem and mesh independence

The Kantorovich theorem is the nonlinear analogue of the ‘stability and consistency imply convergence’ results in differential equations (LeVeque 2007). The simplest example will illustrate the idea for linear problems. Suppose that $u^*$ is the solution of boundary value problem

$$-u''(x) = f(x), u(0) = u(1) = 0$$

for some twice continuously differentiable function $f$. Let $\mathbf{D}^h$ be the standard second-order approximation,

$$\mathbf{D}^h = \frac{1}{h^2}\begin{pmatrix} 2 & -1 & 0 & \dots & 0, & 0 \\ -1 & 2 & -1 & ,0 & \dots & 0 \\ 0 & -1 & 2 & -1, & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots, & ,0, & -1 & 2 & -1 \\ 0 & \dots, & \dots,, & 0 & -1 & 2 \end{pmatrix},$$

where $h = 1/(N+1)$ is the spatial mesh width. Define $\mathcal{E}^h : C[0,1] \to \mathbb{R}^N$ by

$$\mathcal{E}^h(u)_i = u(x_i),$$

where $x_i = (i+1)h$ is the $i$th interior grid point.

Second-order consistency is the statement that

$$A^h \mathcal{E} u^* - \mathcal{E} f = O(h^2).$$

Stability is the uniform boundedness of $\|(\mathbf{D}^h)^{-1}\|$, which is easy to verify. Together, stability and consistency imply that

$$\|\mathcal{E} u^* - (\mathbf{D}^h)^{-1} \mathcal{E} f l\| = O(h^2)$$

uniformly in $h$. Hence, if $\mathbf{u}^h = (\mathbf{D}^h)^{-1} \mathcal{E} f$ then

$$\|\mathbf{u}^h - \mathcal{E} u^*\| = O(h^2),$$

which is second-order convergence.

### *3.1. The theorem*

The Kantorovich theorem (Kantorovich and Akilov 1982) is a convergence result for Newton's method that replaces the standard assumption that there is a solution with the statement that $\mathbf{F}(\mathbf{x}_0)$ is sufficiently small (consistency). That coupled with a uniform bound on $\mathbf{F}'(\mathbf{x})^{-1}$ in a neighbourhood of $\mathbf{x}_0$ (stability) *implies* that there is a unique solution $\mathbf{x}^*$ in a (smaller) neighbourhood of $\mathbf{x}_0$ and that $\mathbf{x}^*$ is near $\mathbf{x}_0$ (convergence!).

We use the formulation of the theorem from Ortega and Rheinboldt (1970). There are several variations of the theorem. In addition to Ortega and Rheinboldt (1970) and Kantorovich and Akilov (1982), one can find examples of alternative formulations in Kelley (1995), Dennis (1969), and Dennis (1971). In Section 6.2 we will present a version from Qi and Sun (1993) for non-smooth functions.

The assumptions are similar to the standard assumptions.

**Assumption 3.1.** $\mathbf{F}$ is defined and Lipschitz continuously differentiable in $\mathcal{D} \subset \mathbb{R}^N$.

- Equation (2.1) holds for all $x, y \in \mathcal{D}$.
- $\|\mathbf{F}'(\mathbf{x})^{-1}\| \le \beta$ for all $\mathbf{x} \in \mathcal{D}$.
- There is $\mathbf{x}_0 \in \mathcal{D}$ such that $\|\mathbf{F}'(\mathbf{x}_0)^{-1}\mathbf{F}(\mathbf{x}_0)\| \le \eta$.
- $\alpha \equiv \beta\gamma\eta \le 1/2$.
- The ball $\mathcal{B}(\mathbf{x}_0, r_+) \subset \mathcal{D}$, where

$$r_\pm = \frac{1 \pm \sqrt{1 - 2\alpha}}{\beta\gamma}.$$

Note that the assumptions on $\mathbf{x}_0$ cannot be stated in terms of $\|\mathbf{x}_0 - \mathbf{x}^*\|$ because the existence of $\mathbf{x}^*$ is part of the conclusion of the theorem. Rather, one uses $\|\mathbf{F}'(\mathbf{x}_0)^{-1}\mathbf{F}(\mathbf{x}_0)\|$ as a surrogate. However, the theorem will imply that

$$\eta \ge \|\mathbf{F}'(\mathbf{x}_0)^{-1}\mathbf{F}(\mathbf{x}_0)\| = \|\mathbf{e}_0\| + O(\|\mathbf{e}_0\|^2).$$

Since $\beta = \|\mathbf{F}'(\mathbf{x}^*)^{-1}\| + O(\|\mathbf{e}_0\|)$, the assumption $\beta\gamma\eta < 1/2$ implies that

$$\|\mathbf{e}_0\| \leq \frac{\|\mathbf{F}'(\mathbf{x}^*)^{-1}\|^{-1}}{2\gamma} + O(\|\mathbf{e}_0\|^2),$$

which is close to (2.3) if $\|\mathbf{e}_0\|$ is small enough to permit one to neglect the second-order term. One remarkable fact about the Kantorovich theorem is that there is no need for $\|\mathbf{e}_0\|$ to be small to obtain convergence. In that sense the theorem is a global convergence result. This is why one cannot expect q-quadratic convergence for the entire iteration. The r-quadratic convergence one obtains from the theorem is remarkable.

**Theorem 3.2.** Let Assumption 3.1 hold. Then there is a unique root $x^*$ of $F$ in $\mathcal{B}(\mathbf{x}_0, r_+)$, the Newton iteration with $\mathbf{x}_0$ as the initial iterate converges to $\mathbf{x}^*$, and $x_n \in \mathcal{B}(\mathbf{x}_0, r_-)$ for all $n$. The errors satisfy the estimate

$$\|\mathbf{e}_n\| \leq \frac{(2\beta\eta\gamma)^{2^n}}{2^n\beta\gamma}, \tag{3.1}$$

and hence the convergence is r-quadratic.

We refer to Ortega and Rheinboldt (1970) for the proof and to Kelley (1995) for a simpler proof of an analogous theorem for the chord method. The proof in Kelley (1995) is based on directly showing that the chord map

$$\mathbf{G}_C(\mathbf{x}) = \mathbf{x} - \mathbf{F}'(\mathbf{x}_0)^{-1}\mathbf{F}(\mathbf{x})$$

is a contraction on $\mathcal{B}(\mathbf{x}_0, r_-)$. The proof of Theorem 3.2 uses a recursion relation for a bound on the steps to show that the Newton map

$$\mathbf{G}_N(\mathbf{x}) = \mathbf{x} - \mathbf{F}'(\mathbf{x})^{-1}\mathbf{F}(\mathbf{x})$$

is a contraction and to obtain the convergence rate estimate (3.1).

### *3.2. A boundary value problem*

In this section we will illustrate in some detail how the Kantorovich theorem can be applied to show convergence of a finite-difference approximation to a nonlinear two-point boundary value problem. We will begin with an existence/uniqueness result for the continuous problem and then directly estimate $\eta$, $\beta$ and $\gamma$ to apply the Kantorovich theorem.

The boundary problem is

$$-u''(x) - \cos(u) = 0, \quad u(0) = u(1) = 0. \tag{3.2}$$

We will use the contraction mapping theorem in the space $C[0,1]$ to prove existence and uniqueness of a solution of (3.2).

Let $\mathcal{G}$ be the inverse of the negative Laplacian in one space dimension with homogeneous Dirichlet boundary conditions, that is,

$$\mathcal{G}(u)(x) = \int_0^1 g(x,y)u(y)\,\mathrm{d}y, \tag{3.3}$$

where the Green's function $g$ is

$$g(x,z) = \begin{cases} x(1-z) & 0 < x < z, \\ z(1-x) & z < x < 1. \end{cases}$$

If $f \in C[0,1]$ and $u = \mathcal{G}(f)$, then

$$-u'' = f, \quad u(0) = u(1) = 0.$$

Hence (3.2) has a solution if and only if the equivalent integral equation

$$\mathcal{F}(u)(x) = u(x) - \int_0^1 g(x,y)\cos(u(y))\,\mathrm{d}y = 0 \tag{3.4}$$

has a solution. We express the integral equation as a fixed-point problem

$$u(x) = \mathcal{K}(u)(x) \equiv \mathcal{G}(\cos(u))(x),$$

and will show that $\mathcal{K}$ is a contraction on $C[0,1]$.

Clearly

$$|\cos(u) - \cos(v)| \le |u - v|, \quad \text{for all } u, v.$$

So

$$|\mathcal{K}(u)(x) - \mathcal{K}(v)(x)| \le \|u - v\|_\infty \int_0^1 g(x,y)\,\mathrm{d}y.$$

Note that

$$w(x) = \int_0^1 g(x,y)\,\mathrm{d}y = \frac{x(1-x)}{2}$$

and hence $\|w\|_\infty = 1/8$. Thus

$$\|\mathcal{K}(u) - \mathcal{K}(v)\|_\infty \le \|u - v\|_\infty / 8.$$

So $\mathcal{K}$ is a contraction on $C[0,1]$. We now know that the boundary value problem has a unique solution $u^* \in C[0,1]$. We have also derived a uniform bound for $\mathcal{F}'(u)^{-1}$. For any $w \in C[0,1]$ we have (using (2.27))

$$\mathcal{F}'(u)w(x) = w(x) + \int_0^1 g(x,y)\sin(u(y))w(y)\,\mathrm{d}y$$

and so

$$\|\mathcal{F}'(u)w\|_\infty \ge (7/8)\|w\|_\infty.$$

Hence

$$\|\mathcal{F}'(u)^{-1}\|_\infty \leq 8/7$$

for all $u \in C[0,1]$. We can also conclude from the above that $\mathcal{F}'$ is Lipschitz continuous with Lipschitz constant $\gamma \leq 1/8$.

We discretize the problem with central finite differences with a uniform mesh width $h$. The discrete problem is

$$\mathbf{D}^h\mathbf{u}^h + \cos(\mathbf{u}^h) = 0.$$

We precondition with $\mathbf{G}^h \equiv (\mathbf{D}^h)^{-1}$ and obtain the analogue of (3.4):

$$\mathbf{F}^h(\mathbf{u}^h) = \mathbf{u}^h + \mathbf{G}^h\cos(\mathbf{u}^h) \equiv \mathbf{u}^h + \mathbf{K}^h(\mathbf{u}^h). \tag{3.5}$$

The smoothness of $u^*$ implies that there is $\eta_0 > 0$ such that

$$\|\mathbf{F}^h(\mathcal{E}^h u^*)\|_\infty \leq \eta_0 h^2 \tag{3.6}$$

and that

$$\|(\mathbf{F}^h)^{-1}(u)\|_\infty \leq 2 \tag{3.7}$$

uniformly in $u$ and $h$. Clearly the Lipschitz constant $\gamma^h$ of of $(\mathbf{F}^h)'$ is $\leq 1/4$ for $h$ sufficiently small.

We are can now apply the Kantorovich theorem to show that a solution $\mathbf{u}^h$ of (3.5) exists and that

$$\|\mathbf{u}^h - \mathcal{E}^h u^*\|_\infty = O(h^2). \tag{3.8}$$

We will apply the theorem with $\beta = 2$ from (3.7), $\eta = 2\eta_0 h^2$ from (3.6) and (3.7), and $\gamma = \gamma^h = 1/4$. Then

$$\alpha = \eta_0 h^2 < 1/2$$

for $h$ sufficiently small. So there is a solution $\mathbf{u}^h$ in $\mathcal{B}(\mathcal{E}^h u^*, r_-)$ where

$$r_- = \frac{1-\sqrt{1-2\alpha}}{\beta\gamma} = \frac{1-\sqrt{1-2\eta_0 h^2}}{1/2} = O(h^2).$$

Summarizing, stability (3.7) and consistency (3.6) imply convergence.

### *3.3. Mesh independence results*

The example in Section 3.2 made use of the fact that the parameters in the Kantorovich theorem could be estimated independently of the mesh $h$. Hence the convergence estimate in (3.1) is also independent of $h$. This is a simple example of a *mesh independence* theorem. In this section we will explore two variations of this idea. Classical mesh independence results are about the convergence of the iteration statistics to those of an underlying infinite-dimensional problem as a grid is refined. In this case the errors are deterministic and we look at some of the ideas in this section. After that we consider tracking theorems. These results describe convergence of the iteration statistics when the errors are stochastic and depend, for example, on the sample size.

### *3.3.1. Deterministic errors*

The concept of mesh independence has its origin in Allgower, Böhmer, Potra and Rheinboldt (1986). While the results may seem obvious, mesh independence is a very useful concept in both nonlinear equations (Kelley and Sachs 1991, Ferng and Kelley 2000) and optimization (Sachs 1990, Hintermüller and Ulbrich 2003).

The idea is that one is approximating an infinite-dimensional equation

$$\mathcal{F}(u) = 0, \tag{3.9}$$

defined on a Banach space $X$, with a sequence of finite-dimensional problems

$$\mathbf{F}^h(\mathbf{u}) = 0 \tag{3.10}$$

on $\mathbb{R}^N$. The boundary value problem from Section 3.2 and the Chandrasekhar $H$-equation from Section 2.10 are examples of this situation. We will be interested in a solution $u^*$ of (3.9) at which the standard assumptions hold.

In both of these examples the performance of Newton's method is mesh-independent in the sense that the relative residuals not only converge to zero, but also converge for each iteration as the mesh is refined. In Section 3.2 we introduced one way to quantify this. Let $\mathcal{E}^h : X \to \mathbb{R}^N$ be the projection from the space $X$ to $\mathbb{R}^N$ that encodes the discretization. For a finite-difference approximation, for example, $\mathcal{E}^h$ could be evaluation at the grid points.

Convergence of the approximation means that the solution $\mathbf{u}^h$ of (3.10) satisfies

$$\lim_{h\to 0} \|\mathbf{u}^h - \mathcal{E}^h u^*\| = 0. \tag{3.11}$$

In many cases, including the examples in Sections 3.2 and 2.10, one can show that the Newton iterations also converge in the sense that

$$\lim_{h\to 0} \|\mathbf{u}_n^h - \mathcal{E}^h u_n\| = 0, \tag{3.12}$$

where $\mathbf{u}_n^h$ is the $n$th Newton iteration for (3.10) and $u_n$ is the $n$th Newton iteration for (3.9).

We will express the fact that the iteration statistics converge in two ways, both of which follow from (3.11) and (3.12). These results also hold for the JFNK methods we discussed in Section 2.6. We will denote the infinite-dimensional iteration

$$u_{n+1} = u_n - \mathcal{F}'(u_n)^{-1}\mathcal{F}(u_n).$$

- Let $\epsilon > 0$ and let $k^h$ be the least $k$ such that $\|\mathbf{F}^h(\mathbf{u}_k^h)\|/\|\mathbf{F}^h(\mathbf{u}_0^h)\| < \epsilon$. Let $k^0$ be the least $k$ such that
$$\|\mathcal{F}(u_k)\|/\|\mathcal{F}(u_0)\| < \epsilon.$$
Then, for all $h$ sufficiently small, $|k^h - k^0| \leq 1$.
- Let $K > 0$ and let $\epsilon > 0$ be given. Then, for $h$ sufficiently small,
$$\left| \frac{\|\mathbf{F}^h(\mathbf{u}_k^h)\|}{\|\mathbf{F}^h(\mathbf{u}_0^h)\|} - \frac{\|\mathcal{F}(u_k^h)\|}{\|\mathcal{F}(u_0)\|} \right| < \epsilon$$
for all $0 \leq k \leq K$.

### *3.3.2. Tracking theorems*

The results in Willert, Chen and Kelley (2015) consider the case where the function $\mathbf{F}$, the Jacobian $\mathbf{F}'$ and Jacobian-vector products are not evaluated directly, but are approximated using internal Monte Carlo computations. The work was motivated by problems in neutron transport (Willert, Kelley, Knoll and Park 2013, Knoll, Park and Smith 2011), where a Monte Carlo simulation was embedded in the residual. The results are similar to mesh independence theorems in that one seeks to show that some finite subset of the iterations converges, but does not seek to drive the approximation error (or its variance) to zero. However, the assumptions and convergence theorems are more technical. We used the term 'tracking' in Willert *et al.* (2015) rather than convergence.

We approximate functions, Jacobians, and Jacobian-vector projects with a Monte Carlo simulation with a sample size $N_{\mathrm{MC}}$. The notation from Willert *et al.* (2015) is as follows.

- $N_{\mathrm{MC}}$ is the sample size for the function and $N_{\mathrm{MC}}^J$ is the sample size for the Jacobian or Jacobian-vector product.
- $\tilde{\mathbf{F}}(\mathbf{x}, N_{\mathrm{MC}})$ is an outcome of the simulation for the residual $\mathbf{F}(\mathbf{x})$.
- $\mathbf{J}(\mathbf{x}, N_{\mathrm{MC}}^J)$ is an outcome of the simulation for the Jacobian $\mathbf{F}'(\mathbf{x})$.
- $\mathbf{J}_p(\mathbf{x}, \mathbf{v}, N_{\mathrm{MC}}^J)$ is an outcome of the simulation for the Jacobian-vector product $\mathbf{F}'(\mathbf{x})\mathbf{v}$.

We assume that the evaluations of $\tilde{\mathbf{F}}$, $\mathbf{J}$ and $\mathbf{J}_p$ are independent.

Suppose the standard assumptions hold and that Newton's method converges for all $\mathbf{u} \in \mathcal{B}(\mathbf{x}^*, \rho)$. We make a consistency assumption for the function and Jacobian evaluations.

**Assumption 3.3.** There are functions $c_F$, $c_J$ and $c_{Jv}$ and an open set $\mathcal{B}'$ which contains $\mathcal{B}(\mathbf{x}^*, \rho)$ such that, for all $\mathbf{x} \in \mathcal{B}'$, unit vectors $\mathbf{v} \in \mathbb{R}^N$ and

$\delta > 0$,

$$\mathrm{Prob}\left( \|\mathbf{F}(\mathbf{x}) - \tilde{\mathbf{F}}(\mathbf{x}, N_{\mathrm{MC}})\| > \frac{c_F(\delta)}{\sqrt{N_{\mathrm{MC}}}} \right) < \delta, \tag{3.13}$$

$$\mathrm{Prob}\left( \|\mathbf{F}'(\mathbf{x}) - \mathbf{J}(\mathbf{x}, N_{\mathrm{MC}}^J)\| > \frac{c_J(\delta)}{\sqrt{N_{\mathrm{MC}}^J}} \right) < \delta. \tag{3.14}$$

In this article we will only summarize the results from Willert *et al.* (2015) on methods which use full Jacobians. The assumptions on the quality of the residual evaluation and the Jacobian are not sufficient to guarantee quadratic convergence. The algorithm increases the number of samples as the iteration progresses, which sometimes reflects practice. One would never increase the number of samples rapidly enough to capture superlinear convergence, and the theory reflects that. The algorithm from Willert *et al.* (2015) is given in algorithm **newton_MC**.

---

```
newton_MC($u, N_{\mathrm{MC}}, N_{\mathrm{MC}}^J, N_{\mathrm{inc}}, \hat{\eta}, \tau_r, \tau_a$)
  Evaluate $\mathbf{r}_{\mathrm{MC}} = \tilde{\mathbf{F}}(u, N_{\mathrm{MC}})$; $\tau \leftarrow \tau_r\|\mathbf{r}_{\mathrm{MC}}\| + \tau_a$.
  while $\|\mathbf{r}_{\mathrm{MC}}\| > \tau$ do
    Compute $\mathbf{J}(\mathbf{x}, N_{\mathrm{MC}}^J)$
    Find $\mathbf{s}$ which satisfies $\|\mathbf{J}(u, N_{\mathrm{MC}}^J)\mathbf{s} + \tilde{\mathbf{F}}(\mathbf{x}, N_{\mathrm{MC}})\| \le \eta\|R_{\mathrm{MC}}\|$ with $0 \le \eta \le \hat{\eta}$
    $\mathbf{x} \leftarrow \mathbf{x} + \mathbf{s}$
    Evaluate $\mathbf{r}_{\mathrm{MC}} = \tilde{\mathbf{F}}(\mathbf{x}, N_{\mathrm{MC}})$;
    $N_{\mathrm{MC}} \leftarrow N_{\mathrm{inc}} N_{\mathrm{MC}}$
  end while
```

---

**Theorem 3.4.** Let (3.13) and (3.14) from Assumption 3.3 and the assumptions of Theorem 2.2 hold. Let $r_{\mathrm{Newton}} \in (0,1)$ be given and assume that $\|\mathbf{e}_0\| \le \rho$ and $\hat{\eta}$ are small enough for the inexact Newton iteration to converge q-linearly with a q-factor $r_{\mathrm{Newton}}$. Let a positive integer $K$, $r \in (r_{\mathrm{Newton}}, 1)$ and $\omega \in (0,1)$ be given. Then there are $\hat{\eta}$, $N_{\mathrm{MC}}$, $N_{\mathrm{MC}}^J$ and $N_{\mathrm{inc}}$, such that, with probability $(1-\omega)$ for all $1 \le n \le K$, the iteration produced by algorithm **newton_MC** satisfies

$$\|\mathbf{e}_n\| \le r^n \|\mathbf{e}_0\|, \tag{3.15}$$

If one approximates matrix–vector products rather than the Jacobian itself, then assumption (3.14) must be replaced by one on the matrix–vector product computation. The subtle, and important, problem with this is that there is no underlying matrix. This can (and does: see Willert *et al.* 2013) produce error accumulation in the Krylov method. Simoncini and Szyld

(2003*a*, 2003*b*, 2007) explain this in detail. Willert *et al.* (2015) use that analysis to derive very technical tracking results, which we mercifully omit from this paper, for matrix-free methods.

## 4. Pseudo-arclength continuation

This section is the first of three (see also Sections 6.4 and 7) on continuation methods. In this section we will look at parameter-dependent nonlinear equations

$$\mathbf{F}(\mathbf{x}, \lambda) = 0 \tag{4.1}$$

and the dependence of the solution(s) on the parameter $\lambda$. The parameter $\omega$ in the $H$-equation is one example. Other examples are the load in a mechanics problem or the voltage in circuit design. In general $\lambda$ can be a vector, but we will only consider scalars in this section.

The study of parameter-dependent systems is deeply connected to dynamics (Govaerts 2000, Keller 1987, Marsden and McCracken 1976, Kuznetsov 1998), but we will only examine that connection in a superficial way in Section 7. In particular we will not cover bifurcation, the case where two different solution paths intersect. One must use the higher derivative tensors to understand singularities of that type.

One may think that this is a trivial problem. One could begin with $\lambda_0$ and solve (4.1) for $\lambda = \lambda_0$ with the Armijo line search from Section 2.7 to find $\mathbf{x}^*(\lambda_0)$. Then one picks an increment $\delta_\lambda$ in the parameter and solves (4.1) for $\mathbf{x}(\lambda+\delta_\lambda)$ with $\mathbf{x}^*(\lambda)$ as the initial iterate. Algorithm **simple_continuation** is a formal description of this simple parameter continuation idea.

```
simple_continuation($\mathbf{x}, \lambda, \delta_\lambda, \lambda_{\max}, \mathbf{F}, \tau_a, \tau_r$)
    Compute $\mathbf{x}^*(\lambda)$ with algorithm newton_armijo with $\mathbf{x}$ as the initial iterate.
    while $\lambda \le \lambda_{\max}$ do
        $\lambda \leftarrow \lambda + \delta_\lambda$
        Compute $\mathbf{x}^*(\lambda)$ with algorithm newton_armijo with $\mathbf{x}^*(\lambda - \delta_\lambda)$ as the initial iterate.
    end while
```

Algorithm **simple_continuation** can only succeed if the range of $\lambda$ is infinite. In the case of the $H$-equation, for example, there are no real solutions for $\omega > 1$, and something must go wrong when $\omega = 1$ and the Jacobian is singular at the solution. We explain that in the next section and then describe one possibility for resolving the problem.

### *4.1. The implicit function theorem*

The implicit function theorem says that algorithm **simple_continuation** will successfully follow the solution path as long as $\mathbf{F}'$ is safely non-singular and also provides an estimate of $\delta_\lambda$. We give a proof that is a nice application of the Kantorovich theorem.

We begin with some notation. If $\mathbf{F}(\mathbf{x},\lambda)$ is a function of $\mathbf{x} \in \mathbb{R}^N$ and $\lambda \in R$, then $\mathbf{F}_\mathbf{x}$ will denote the Jacobian in the $\mathbf{x}$ variable. Hence $\mathbf{F}_\mathbf{x}$ is an $N \times N$ matrix. Similarly $\mathbf{F}_\lambda \in \mathbb{R}^N$ is the partial of $\mathbf{F}$ with respect to the scalar variable $\lambda$.

**Theorem 4.1.** Assume the following:

- $\mathbf{F}(\mathbf{x}_0,\lambda_0) = 0$;
- $\mathbf{F}$ is a continuously differentiable function of $\lambda$;
- $\mathbf{F}_\mathbf{x}$ and $\mathbf{F}$ are Lipschitz continuous in $(\mathbf{x},\lambda)$ with Lipschitz constant $\gamma$, that is,
$$\|\mathbf{F}(\mathbf{x},\lambda) - \mathbf{F}(\mathbf{y},\mu)\| \le \gamma(\|\mathbf{x}-\mathbf{y}\| + |\lambda-\mu|)$$
and
$$\|\mathbf{F}_\mathbf{x}(\mathbf{x},\lambda) - \mathbf{F}_\mathbf{x}(\mathbf{y},\mu)\| \le \gamma(\|\mathbf{x}-\mathbf{y}\| + |\lambda-\mu|);$$
- $\mathbf{F}_\mathbf{x}(\mathbf{x}_0,\lambda_0)$ is non-singular and $\|\mathbf{F}_\mathbf{x}^{-1}(\mathbf{x}_0,\lambda_0)\| \le \beta_0$.

Then there are $\Delta$ and $r$, which depend only on $\beta$ and $\gamma$, such that the following hold:

- there is a solution $\mathbf{x}(\lambda)$ of (4.1) for all $\lambda$ such that $|\lambda-\lambda_0| \le \Delta$;
- $\mathbf{x}(\lambda_0) = \mathbf{x}_0$;
- $\mathbf{x}(\lambda)$ is the only solution of (4.1) for $\|\mathbf{x}-\mathbf{x}_0\| \le r$;
- $\mathbf{x}(\lambda)$ is a continuously differentiable function of $\lambda$.

*Proof.* Similarly to the proof of Theorem 2.2, Lipschitz continuity of $\mathbf{F}_\mathbf{x}$ implies that $\mathbf{F}_\mathbf{x}(\mathbf{x},\lambda)$ is non-singular in the set

$$\mathcal{D} = \left\{ (\mathbf{x},\lambda) \mid \|\mathbf{x}-\mathbf{x}_0\| + |\lambda-\lambda_0| \le \frac{1}{2\gamma\|\mathbf{F}_\mathbf{x}^{-1}(\mathbf{x}_0,\lambda_0)\|} \right\},$$

and that

$$\|\mathbf{F}_\mathbf{x}^{-1}(\mathbf{x},\lambda)\| \le \beta \equiv 2\beta_0$$

for all $(\mathbf{x},\lambda) \in \mathcal{D}$.

We have $\beta$ and $\gamma$ in hand for an application of the Kantorovich theorem. We use Lipschitz continuity of $\mathbf{F}$ to obtain

$$\|\mathbf{F}(\mathbf{x}_0,\lambda)\| = \|\mathbf{F}(\mathbf{x}_0,\lambda) - \mathbf{F}(\mathbf{x}_0,\lambda_0)\| \le \gamma|\lambda-\lambda_0| \equiv \eta.$$

So if $\gamma^2\beta\Delta < 1/2$ and $|\lambda - \lambda_0| \leq \Delta$, we can apply the Kantorovich theorem with

$$\alpha = \beta\gamma\eta = \beta\gamma^2\Delta \leq 1/2,$$

to complete the proof of existence with

$$r = \frac{1 \pm \sqrt{1+2\alpha}}{\beta\gamma}.$$

To show differentiability, we formally differentiate (4.1) with respect to $\lambda$ and note that $\mathbf{x}'(\lambda)$ is the solution $\mathbf{z}$ of

$$\mathbf{F}_{\mathbf{x}}(\mathbf{x}, \lambda)\mathbf{z} = -\mathbf{F}_{\lambda}(\mathbf{x}, \lambda). \qquad \square$$

If $\mathbf{F}$ has higher-order derivatives, one can show that $\mathbf{x}$ has as many derivatives in $\lambda$ as $\mathbf{F}$ does in $(\mathbf{x}, \lambda)$ (Rabinowitz 1971).

If $\mathbf{F}_{\mathbf{x}}(\mathbf{x}^*, \lambda^*)$ is singular, then the implicit function cannot be used to assert that there are solutions near $\lambda^*$. In the case of the $H$-equation, $\mathbf{F}_{\mathbf{h}}(\mathbf{h}^*, 1)$ is singular and (2.34) implies that there is no real solution for $\omega > 1$.

So, does the solution arc stop dead at the point $(H(1), 1)$? The answer is that for many common singularities, including the one for the $H$-equation, the solution arc does not terminate abruptly but either loops back or becomes unbounded (Crandall and Rabinowitz 1971, Rabinowitz 1971, Keller 1987). We will discuss some specific examples in the rest of this section.

### *4.2. Simple folds and pseudo-arclength continuation*

We will begin this section with an example that is both simple and general. Consider the scalar equation

$$f(x, \lambda) = x^2 - \lambda.$$

The function $f$ is Lipschitz continuous if $x$ is restricted to a bounded set. Since $f_x = 2x$ the Lipschitz constant of the derivative is 2. There is a unique solution $x(0) = 0$ when $\lambda = 0$, no real solution when $\lambda < 0$, and two solutions when $\lambda > 0$. The singularity of $f_x$ at $\lambda = 0$ has, as we will see, exactly the same structure as the singularity of the $H$-equation when $\omega = 1$.

Suppose we begin with $(x, \lambda) = (1, 1)$ and try to use simple continuation to reach and pass $\lambda = 0$. The continuation will fail because there is no solution for $\lambda < 0$. However, the path of solutions continues beyond $\lambda = 0$, the only difference being that the sign of $x$ changes. How can we modify the simple continuation algorithm to follow the path without getting stuck at the singularity?

One way would be to interchange the roles of $\lambda$ and $x$. This is the idea of the PITCON (Rheinboldt 1986) code. If we do that, then $f_\lambda = -1$ is never singular and we can continue with ease. The problem with this approach is that one must identify the variable to exchange with $\lambda$. Rheinboldt (1986) identifies many situations where this is readily done. There are many advanced continuation codes (Salinger *et al.* 2002, Doedel and Kernévez 1986, Govaerts 2000) and some very good books (Govaerts 2000, Kuznetsov 1998, Doedel 1997, Keller 1987, Rheinboldt 1986) on the topic.

The approach we describe in this section, pseudo-arclength continuation, attempts to parametrize the solution arc and then add an approximation to arclength as a new parameter (Keller 1987).

To see how this would work, suppose that $x(s)$ and $\lambda(s)$ are functions of an arclength parameter $s$. Setting $\dot{x} = \mathrm{d}x/\mathrm{d}s$ and $\dot{\lambda} = \mathrm{d}\lambda/\mathrm{d}s$, we use the arclength normalization

$$\dot{x}^2 + \dot{\lambda}^2 = 1. \tag{4.2}$$

We define an expanded equation in $\mathbf{z} = (x, \lambda)^T$ with $s$ as the parameter:

$$\mathbf{G}(\mathbf{z}, s) \equiv \begin{pmatrix} f(x,\lambda) \\ \dot{x}^2 + \dot{\lambda}^2 - 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \tag{4.3}$$

If $\mathbf{g}_\mathbf{z}$ is non-singular and one can effectively approximate $\dot{x}$ and $\dot{\lambda}$, then one can apply simple continuation to the expanded system.

We will investigate the non-singularity of $\mathbf{G}_\mathbf{z}$ first. Clearly

$$\mathbf{G}_\mathbf{z} = \begin{pmatrix} f_x & f_\lambda \\ \dot{x} & \dot{\lambda} \end{pmatrix} = \begin{pmatrix} 2x & -1 \\ \dot{x} & \dot{\lambda} \end{pmatrix},$$

so

$$\det(\mathbf{G}_\mathbf{z}) = 2x\dot{\lambda} + \dot{x}.$$

Use $f(x,\lambda) = 0$ to get

$$0 = \mathrm{d}f(x,\lambda)/\,\mathrm{d}s = f_x\dot{x} + f_\lambda\dot{\lambda} = 2x\dot{x} - \dot{\lambda}$$

and so $2x\dot{x} = \dot{\lambda}$

Now multiply $\det(\mathbf{G}_\mathbf{z}) = 2x\dot{\lambda} + \dot{x}$ by $\dot{x}$ and

$$\dot{x}\det(\mathbf{G}_\mathbf{z}) = 2x\dot{x}\dot{\lambda} + \dot{x}^2.$$

Use $2x\dot{x} = \dot{\lambda}$ and

$$\dot{x}\det(\mathbf{G}_\mathbf{z}) = \dot{\lambda}^2 + \dot{x}^2 = 1.$$

So $\det(\mathbf{G}_\mathbf{z}) \neq 0$.

One way to approximate $\dot{x}$ and $\dot{\lambda}$ is to use simple continuation for two values of $\lambda$ and then switch to continuation in $s$. If one has two solutions $(x(\lambda_0), \lambda_0)$ and $(x(\lambda_{-1}), \lambda_{-1})$ in hand, then one could approximate $ds$ by using

$$\dot{x}^2 + \dot{\lambda}^2 = 1$$

to conclude that if

$$(x_0 - x_{-1})^2 + (\lambda_0 - \lambda_{-1})^2 \equiv ds^2$$

then

$$\dot{x}_0 \approx \frac{x_0 - x_{-1}}{ds} \quad \text{and} \quad \dot{\lambda}_0 \approx \frac{\lambda_0 - \lambda_{-1}}{ds}.$$

Having $\dot{x}_0$ and $\dot{\lambda}_0$, we can use any increment $ds$ in $s$ we like, and replace $\dot{x}^2 + \dot{\lambda}^2 - 1 = 0$ with

$$\dot{x}_0(x - x_0) + \dot{\lambda}_0(\lambda - \lambda_0) - ds = 0.$$

So, the equation $\mathbf{G}(\mathbf{z}, s) = 0$ changes as we increment $s$ because the approximation of $\dot{z}$ depends on the current point in the path. However, for sufficiently small $ds$ this approximation works very well, and the implicit function theorem applies at every stage of the continuation. The analysis is a bit trickier in several variables, but this simple problem captures all the essential ideas.

In several variables we must require that either $\mathbf{F}_\mathbf{x}$ be non-singular or that the singularity be a *simple fold*.

**Definition 4.2.** The point $(\mathbf{x}^*, \lambda^*)$ is a simple fold point if:

- $\mathbf{F}(\mathbf{x}^* \lambda^*) = 0$;
- $\mathbf{F}_\mathbf{x}(\mathbf{x}^* \lambda^*)$ has a one-dimensional null space;
- $\mathbf{F}_\lambda(\mathbf{x}^* \lambda^*)$ is not in the range of $\mathbf{F}_\mathbf{x}(\mathbf{x}^* \lambda^*)$.

For our simple scalar example, the first two conditions in the definition are trivially true. Since $f_x(0,0) = 0$ and $f_\lambda(0,0) = -1$, the third condition holds as well.

Pseudo-arclength continuation in the case of several variables proceeds in the same way as our scalar example. We seek to advance in arclength by $ds$ from a point $(\mathbf{x}_0, \lambda_0)$ on the solution path. We do this by solving the expanded system for $\mathbf{z} = (\mathbf{x}^T, \lambda)^T$:

$$\mathbf{G}(\mathbf{z}, s) \equiv \begin{pmatrix} \mathbf{F}(\mathbf{x}, \lambda) \\ \mathbf{N}(\mathbf{z}, s) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \tag{4.4}$$

In (4.4),

$$\mathbf{N}(\mathbf{z}, s) \approx \|\dot{\mathbf{x}}\|^2 + \dot{\lambda}^2 - 1$$

is a normalization term. One common choice is the secant normalization

$$\mathbf{N}(\mathbf{z},s) = \theta\left(\frac{\mathbf{x}(s_0)-\mathbf{x}(s_{-1})}{s_0-s_{-1}}\right)^T(\mathbf{x}(s)-\mathbf{x}(s_0)) + (1-\theta)\left(\frac{\lambda(s_0)-\lambda(s_{-1})}{s_0-s_{-1}}\right)(\lambda(s)-\lambda(s_0)) - (s-s_0), \quad (4.5)$$

which we can use when two points on the path are available. To start the continuation, a typical choice is the norm-based normalization

$$\mathbf{N}(\mathbf{z},s) = \theta\|\mathbf{x}-\mathbf{x}(s_0)\|^2 + (1-\theta)|\lambda-\lambda(s_0)|^2 - (s-s_0)^2. \quad (4.6)$$

The parameter $\theta$ is a scaling parameter that balances the size of the vector norm with the scalar parameter $\lambda$.

Summarizing, algorithm **arclength_continuation** is a formal description of pseudo-arclength continuation.

```
arclength_continuation($\mathbf{x}, \lambda, ds, S, \mathbf{F}, \tau_a, \tau_r$)
  Set $s = 0, \mathbf{x}(0) = \mathbf{x}, \lambda(0) = \lambda$.
  Compute $\mathbf{x}^*(0)$ with algorithm newton_armijo with $\mathbf{x}$ as the initial iterate. You now have $\mathbf{z}^*(0)$.
  while $s \le S$ do
    $s \leftarrow s + ds$.
    Compute $\mathbf{z}^*(s)$ with algorithm newton_armijo with $\mathbf{x}^*(s-ds)$ as the initial iterate.
  end while
```

We describe the continuation method as a theorem.

**Theorem 4.3.** Suppose $\mathbf{F}(\mathbf{x}(s),\lambda(s)) = 0$ for $0 \le s \le S$, and for each $s$ either $\mathbf{F}_\mathbf{x}$ is non-singular or $(\mathbf{x}(s),\lambda(s))$ is a simple fold singularity. Let $\mathbf{G}$ be defined by (4.4) with either (4.5) or (4.6) used as normalization. Then the implicit function theorem applies to $\mathbf{G}(\mathbf{x},s)$ for all $0 \le s \le S$. Moreover, for $ds$ sufficiently small, algorithm **arclength_continuation** will find the points $\mathbf{z}(kds) = (\mathbf{x}(kds)^T, \lambda(kds))^T$ for $0 \le kds \le S + ds$.

Theorem 4.3 asserts that if the parameter is changed from $\lambda$ to $s$, then the singularity has been eliminated and the path of solutions is homeomorphic to a line segment (Crandall and Rabinowitz 1971). In that event, all one has to do in order to follow the path of solutions is to apply algorithm **arclength_continuation**. In addition, one can prove a mesh independence result (Ferng and Kelley 2000) for discretizations of continuous problems.

As you might imagine, there are a few details to resolve. The parameter $\theta$ in (4.5) and (4.6) must be used to maintain mesh independence if one is discretizing a continuous problem. The reason for this is that if one uses a solver which is based on the discrete $\ell^2$-norm, that norm does not converge to the integral $L^2$-norm as the mesh is refined. In the case of the midpoint rule, for example,

$$\int_0^1 f(\mu)\,\mathrm{d}\mu \approx \frac{1}{N}\sum_{i=1}^{N} f(\mu_i),$$

and one would use $\theta = 1/N$, where $N$ is the number of spatial mesh points. Without this the discrete approximation would not be consistent with the continuous problem.

Another important detail is the continuation itself. The theorem limits the range of $s$ for a good reason. In some cases, the $H$-equation being one of them, $\lambda(s) \to \lambda_\infty = 0$ as $s \to \infty$ while $\|\mathbf{x}(s)\| \to \infty$. In this case one needs to reduce $ds$ as the continuation progresses in order to ensure that the solutions stay on the path. Even with a line search, there is no guarantee that the solutions will stay on the path (as opposed to jumping to a different solution branch) if the initial iterate is poor. Hence we limit the range of $s$ in the theorem so that one choice of $ds$ will suffice.

Those readers who are familiar with initial value problems will not be surprised to hear that the initial iterate for the solve step in algorithm **arclength_continuation** is called the *predictor*. The algorithm uses the *trivial predictor*, that is, $\mathbf{z}(s)$ as the initial iterate for $\mathbf{z}(s + ds)$. Since one has an estimate for $\dot{\mathbf{z}}(s)$,

$$\dot{\mathbf{z}}(s) \approx ((\mathbf{x}(s) - \mathbf{x}(s - ds))^T, \lambda(s) - \lambda(s - ds)^T/ds,$$

one could use linear extrapolation to form the *linear predictor*

$$\mathbf{z}_0(s + ds) = \mathbf{z}(s) + \dot{\mathbf{z}}(s)ds, \tag{4.7}$$

which generally performs better. Finally, similarly to numerical integration of initial value problems, it is a bad sign if the nonlinear solver requires too many iterations or a line search. Those events are signals that the predictor is poor and that one should reduce $ds$.

In our numerical example we used $ds = 0.002$ and let the continuation run until the line search failed because the limit on stepsize reductions had been exceeded. One would expect that to happen at some point because the solutions are moving farther apart and the predictor's performance is therefore becoming worse as the continuation progresses.

We will illustrate the output for the $H$-equation. In the example we used Newton–GMRES for the nonlinear solver and a forward-difference Jacobian-vector product. We used the forcing term from (2.24). We would expect from (2.34) that there are two solutions for $0 < \omega < 1$, one for each choice of sign for the square root. The continuation computation confirms that.

Typically one plots the progress of a continuation with $\lambda$ on the horizontal axis and a functional of $\mathbf{x}$ on the vertical axis. For the $H$-equation and its

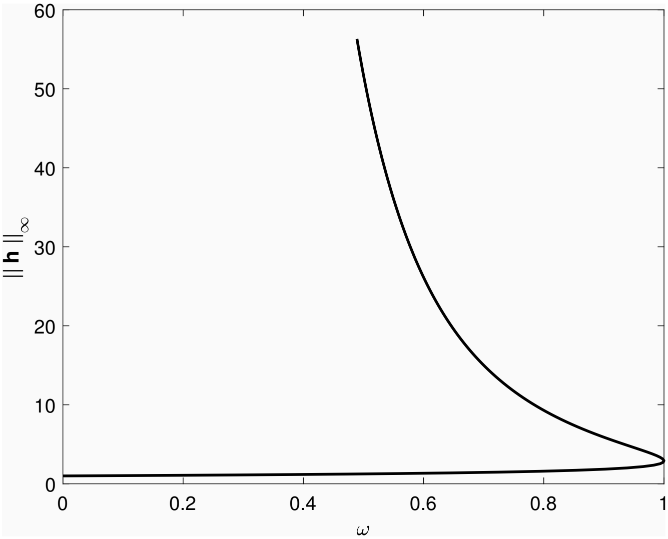

Figure 4.1. Solution path for $H$-equation.

discretization the function is strictly positive (in fact $\geq 1$) and increasing. One could use either $H(1)$ (or $h_N$ in the discrete case) or the $L^1$-norm

$$\int_0^1 H(\mu)\,\mathrm{d}\mu \quad \text{or} \quad \frac{1}{N}\sum_{i=1}^{N} h_i.$$

We use $h_N = \|\mathbf{h}\|_\infty$ to illustrate the steep rise in the size of the solution. See Figure 4.1.

We used the secant normalization for all but the first point on the path and exploit the fact that if $\lambda(0) = 0$ then $H \equiv 1$. This is also true for the discrete case. Hence there is no need to solve the equation at the first point on the path.

## *4.3. The Bratu problem*

In this section we illustrate how multiple solutions of a nonlinear equation can have very different properties when considered as steady-state solutions of a time-dependent problem. Consider the system of ordinary differential equations

$$\dot{\mathbf{x}} = -\mathbf{F}(\mathbf{x}), \tag{4.8}$$

where $\dot{\mathbf{x}} = \mathrm{d}\mathbf{x}/\mathrm{d}t$. We chose the sign of $\mathbf{F}$ to be consistent with the standard practice in pseudo-transient continuation (see Section 7).

A solution $\mathbf{x}^*$ is a *steady-state* solution if it is independent of time. In that case (4.8) implies that

$$0 = \dot{\mathbf{x}}^* = -\mathbf{F}(\mathbf{x}^*).$$

Here $\mathbf{x}^*$ is dynamically stable if the solution $\mathbf{x}$ of the initial value problem

$$\dot{\mathbf{x}} = -\mathbf{F}(\mathbf{x}), \quad \mathbf{x}(0) = \mathbf{x}_0$$

converges to $\mathbf{x}^*$ as $t \to \infty$ for all $\mathbf{x}_0$ sufficiently close to $\mathbf{x}^*$. It is often easier to check for *linear stability*, which is sufficient (but not necessary) for stability. We say $\mathbf{x}^*$ is linearly stable if the eigenvalues of $\mathbf{F}'(\mathbf{x}^*)$ all have positive real part. Not all steady-state solutions are stable, and algorithms such as **newton_armijo** are not aware of dynamics and can (and sometimes do) converge to unstable steady states. We will give an example of this in Section 7.

The *Bratu problem* (Bratu 1914) is an example of this. The steady-state problem is the two-point boundary value problem

$$-u_{xx} = \lambda\,\mathrm{e}^u, \quad u(0) = u(1) = 0. \tag{4.9}$$

We would express the boundary value problem as $-\mathcal{F}(u) = 0$ in the function space. The Fréchet derivative of $\mathcal{F}(u)$ is clearly symmetric and positive definite if $\lambda$ satisfies

$$\pi^2 - \lambda\,\mathrm{e}^u > 0.$$

If $\lambda = 0$, the unique solution of (4.9) is $u \equiv 0$. The Fréchet derivative $\mathcal{F}_u$ at $(u, \lambda) = (0, 0)$ is $-\mathrm{d}^2/\mathrm{d}x^2$ with homogeneous Dirichlet boundary conditions and is positive definite. Hence the implicit function implies that there is a solution of (4.9) for sufficiently small $\lambda > 0$ and that the solution will be linearly stable as long as the smallest eigenvalue of $\mathcal{F}_x$ is positive. As the continuation progresses, the eigenvalue will change sign at the singularity of $\mathcal{F}_x$ and the solution will lose linear stability (if it exists at all). The structure of the Bratu problem (and the $H$-equation as well) implies that the eigenvalue will change sign when one passes the singularity on the continuation path (Keller 1987).

One can solve (4.9) explicitly (Ascher, Mattheij and Russell 1995). The solution is

$$u(x) = -2\ln\left(\frac{\cosh[(x - 1/2)\theta/2]}{\cosh(\theta/4)}\right),$$

where $\theta$ is the solution of the scalar equation

$$\theta = \sqrt{2\lambda}\cosh(\theta/4). \tag{4.10}$$

Similarly to (2.34), (4.10) has two solutions for $0 < \lambda < \lambda^*$ and no real solution for $\lambda > \lambda^* \approx 3.52$. The continuation plot for the Bratu problem is similar to that for the $H$-equation and we invite the reader to do that computation.

## 5. Anderson acceleration

In this section we formulate nonlinear equations as fixed-point problems

$$\mathbf{x} = \mathbf{G}(\mathbf{x}). \tag{5.1}$$

Recall from Section 1.3 that the classic method for solving such problems is Picard iteration:

$$\mathbf{x}_{k+1} = \mathbf{G}(\mathbf{x}_k). \tag{5.2}$$

We gave the well-known theory for Picard iteration in Section 1.3.

Anderson acceleration (Anderson 1965) was designed to accelerate Picard iteration for electronic structure computations. Anderson acceleration differs very little from Pulay mixing (Pulay 1980, Pulay 1982), DIIS (direct inversion on the iterative subspace: Rohwedder and Schneider 2011, Schneider, Rohwedder, Neelov and Blauert 2008, Lin and Yang 2013, Kudin, Scuseria and Cancès 2002) or nonlinear GMRES (Miller 2005, Oosterlee and Washio 2000, Washio and Oosterlee 1997, Carlson and Miller 1998). The results in this section apply to all of these algorithms.

We give an example of the kind of problem for which Anderson acceleration is widely used. The Kohn–Sham equation (Kohn and Sham 1965, Hohenberg and Kohn 1964) in density functional theory (DFT) for electronic structure computations is

$$\mathcal{H}_{ks}[\psi_j] \equiv -\frac{1}{2}\nabla^2\psi_j + V(\rho)\psi_j = \lambda_j\psi_j, \quad j = 1, \ldots, N_e. \tag{5.3}$$

Here $\psi_j$ is the wave function for the $j$th electron of interest, $\mathcal{H}_{ks}$ is the Kohn–Sham Hamiltonian,

$$\rho = \sum_{j=1}^{N} \|\psi_j\|_2^2 \tag{5.4}$$

is the charge density, $N_e$ is the number of electrons of interest, and $V$ is the charge density-dependent potential. It is useful to express (5.3) in vector form:

$$\mathbf{H}(\rho)\Psi = \Lambda\Psi,$$

where $\Psi$ represents the collection of wave functions and $\Lambda$ is a diagonal matrix of eigenvalues. In physics computations $N$ can be several thousand.

Self-consistent field (SCF) iteration begins with an initial iterate for $\rho$; then, using the given $\rho$, one solves the linear eigenvalue problem $\mathbf{H}(\rho)\Psi = \Lambda\Psi$ for the $N_e$ eigenvalues and eigenvectors of interest. One then updates the charge density with (5.4) and continues the iteration until the change in $\rho$ is sufficiently small. This is a fixed-point iteration for the function $\rho$, that is,

$$\rho \leftarrow \mathbf{G}(\rho),$$

which, after discretization, is a fixed-point problem in $\mathbb{R}^{N_F}$. For a real-space formulation, as done in the RMG code (Briggs, Sullivan and Bernholc 1995), $N_F$ would be the number of spatial mesh points if we organized the fixed-point formulation in this way. However, the charge density often converges faster than the wave functions and, when it is important to compute the wave functions, one must formulate the problem in terms of $\Psi$. In that case the size of the problem is the product of the number of spatial mesh points and the number of wave functions $N$. In that case storage can be the limiting factor in a computation.

The problem with applying a version of Newton's method to this problem is that differentiating the output (the collection of wave functions) of the eigencomputation, where eigenvalues of high multiplicity are possible, is difficult in both theory and practice. Therefore, SCF iteration is much more common for large systems; SCF is, of course, Picard iteration, and Anderson acceleration is used in most applications, for example in the Gaussian computational chemistry code (Frisch *et al.* 2009) and in the RMG code (Briggs *et al.* 1995).

Other applications are stiff dislocation dynamics (Gardner *et al.* 2015), fluid–structure interactions (Ganine, Javiya, Hills and Chew 2012), hydrology (Lott, Walker, Woodward and Yang 2012), neutron transport (Willert, Taitano and Knoll 2014, Toth *et al.* 2017), thermal radiation transport (An, Jia and Walker 2017), and multiphysics coupling (Toth 2016, Toth *et al.* 2015, Hamilton *et al.* 2016).

We remind the reader that Anderson acceleration was designed in a context where Newton's method was not practical because obtaining approximate Jacobians or Jacobian-vector products was (and still is) too costly. Comparisons indicate that Newton's method performs better when even moderately accurate derivative information can be had at reasonable cost (Hamilton *et al.* 2016).

### *5.1. Algorithmic description*

```
anderson($\mathbf{x}_0, \mathbf{G}, m$)
    $\mathbf{x}_1 = \mathbf{G}(\mathbf{x}_0)$; $\mathbf{F}_0 = \mathbf{G}(\mathbf{x}_0) - \mathbf{x}_0$
    for $k = 1, \dots$ do
        Choose $m_k \le \min(m, k)$
        $\mathbf{F}(\mathbf{x}_k) = \mathbf{G}(\mathbf{x}_k) - \mathbf{x}_k$
        Minimize $\| \sum_{j=0}^{m_k} \alpha_j^k \mathbf{F}(\mathbf{x}_{k-m_k+j}) \|$ subject to
        $\sum_{j=0}^{m_k} \alpha_j^k = 1$.
        $\mathbf{x}_{k+1} = (1-\beta) \sum_{j=0}^{m_k} \alpha_j^k \mathbf{x}_{k-m_k+j} + \beta \sum_{j=0}^{m_k} \alpha_j^k \mathbf{G}(\mathbf{x}_{k-m_k+j})$
    end for
```

Algorithm **anderson** is a formal description of the method and is the one we will use for analysis. Implementation is a different matter, and there are many examples of efficient implementations (Walker and Ni 2011, Toth and Kelley 2015, Collier *et al.* 2015, Hindmarsh *et al.* 2005, Toth and Pawlowski 2015, Toth 2016).

The parameter $\beta$ is called the mixing parameter. This is the same as the damping parameter in Picard iteration. In many cases one must damp Picard iteration to secure convergence. The damped Picard iteration is

$$\mathbf{x}_{n+1} = (1-\beta)\mathbf{x}_n + \beta\mathbf{G}(\mathbf{x}_n) \equiv \mathbf{G}_\beta(\mathbf{x}_n).$$

Anderson acceleration with mixing parameter $\beta$ is the same as applying the algorithm with $\beta = 1$ to the map $\mathbf{G}_\beta$. Hence there is no loss of generality in setting $\beta \equiv 1$ for analysis.

Anderson maintains a limited history of the iteration of size $m+1$; $m$ is called the *depth*. The iteration uses the most recent $m+1$ residuals $\mathbf{F}(\mathbf{x}_j)$ for $k - m_k \leq j \leq k$ where $m_k \leq \min(k, m)$. The key step in the iteration is solving the *optimization problem*

$$\min \left\| \sum_{j=0}^{m_k} \alpha_j^k \mathbf{F}(\mathbf{x}_{k-m_k+j}) \right\| \quad \text{subject to } \sum_{j=0}^{m_k} \alpha_j^k = 1, \tag{5.5}$$

for the coefficients $\{\alpha_j^k\}$.

Any vector norm can be used in the optimization problem with no change in the theory. The optimization problem is easier to solve if one uses the $\ell^2$-norm, and that is standard practice. In this case the optimization problem for the coefficients can be expressed as a linear least-squares problem and solved very inexpensively. One way to do this is to solve the linear least-squares problem

$$\min \left\| \mathbf{F}(\mathbf{x}_k) - \sum_{j=0}^{m_k-1} \alpha_j^k (\mathbf{F}(\mathbf{x}_{k-m_k+j}) - \mathbf{F}(\mathbf{x}_k)) \right\|_2^2, \tag{5.6}$$

for $\{\alpha_j^k\}_{j=0}^{m_k-1}$. Then one recovers $\alpha_{m_k}^k$ by

$$\alpha_{m_k}^k = 1 - \sum_{j=0}^{m_k-1} \alpha_j^k.$$

Toth and Kelley (2015) point out that other norms could be used. In particular, the optimization problem for the coefficients in either the $\ell^1$- or $\ell^\infty$-norms can be formulated as a linear programming problem and solved directly with many codes (CVX Research 2012). However, the least-squares approach using (5.6) is more efficient, and we will use the $\ell^2$-norm only in this article.

The choice of $m_k$ is, in the original form, simply $\min(m,k)$. One can adapt $m_k$ as the iteration progresses to, for example, enforce well-conditioning of the linear least-squares problem (5.6) (Walker and Ni 2011, Toth 2016, An *et al.* 2017).

One can show (Fang and Saad 2009, Saad, Chelikowsky and Shontz 2010, Rohwedder and Schneider 2011, Walker and Ni 2011, Potra and Engler 2013) that Anderson acceleration is related to multisecant quasi-Newton methods or, in the case of linear problems, GMRES. None of these results lead to a convergence proof, even in the linear case, unless the available storage is large enough to allow GMRES to take a number of iterations equal to the dimension of the problem.

One result from Walker and Ni (2011) illustrates the power of unlimited storage. While not the case seen in practice, this result does illustrate why Anderson acceleration may perform better than Picard iteration in some cases. Unlike the remainder of the results we present in this section, contractivity is not necessary.

**Theorem 5.1.** Let $\mathbf{M}$ be an $N \times N$ matrix with $\mathbf{A} = \mathbf{I} - \mathbf{M}$ non-singular. Let $m \geq N$, $\mathbf{b}$ and $\mathbf{x}_0$ be given. Let $\mathbf{x}_k^G$ be the $k$th GMRES iteration for $\mathbf{A}\mathbf{x} = \mathbf{b}$ with $\mathbf{x}_0$ as the initial iterate. Let $\mathbf{x}_k^a$ be the $k$th Anderson($m$) iteration for $\mathbf{x} = \mathbf{G}(\mathbf{x}) \equiv \mathbf{M}\mathbf{x} + \mathbf{b}$ with $\mathbf{x}_0$ as the initial iterate. Suppose that for some $k > 0$

- $\|\mathbf{A}\mathbf{x}_{k-1}^G - \mathbf{b}\| > 0$ and
- $\|\mathbf{A}\mathbf{x}_{j-1}^G - \mathbf{b}\| > \|\mathbf{A}\mathbf{x}_j^G - \mathbf{b}\|$ for all $0 < j < k$.

Then $\mathbf{x}_{k+1}^A = \mathbf{G}(\mathbf{x}_k^G)$.

This result says that, under most circumstances, Anderson acceleration with $m \geq N$ performs exactly as well as GMRES and that the analysis of preconditioning is the same as for GMRES. GMRES is well known (Nevanlinna 1993, Campbell, Ipsen, Kelley and Meyer 1996*a*, Campbell *et al.* 1996*b*) to converge rapidly for discretizations of second-kind Fredholm integral equations, for example.

### *5.2. Convergence theory*

Toth and Kelley (2015) has the first convergence analysis of the method as used in practice. The central idea in Toth and Kelley (2015) was to show that Anderson acceleration does no harm rather than to prove a general convergence result. The results are consistent with the observations from computational chemistry (Foresman and Frisch 1996). For example, a good initial iterate is needed for convergence in many cases.

### *5.2.1. Linear problems*

We will begin with an analysis of the linear case. The convergence theory, at least as it stands today, only shows that Anderson acceleration does not degrade the convergence of Picard iteration. However, in practice Anderson acceleration is often (but not always) much better. There is at present no satisfactory characterization of problems for which Anderson is better.

**Theorem 5.2.** Let $\mathbf{M}$ be an $N \times N$ matrix with $\|\mathbf{M}\| = c < 1$. Let $m \geq 0$. Then Anderson$(m)$ acceleration, when applied to $\mathbf{G}(\mathbf{x}) = \mathbf{M}\mathbf{x} + \mathbf{b}$, converges to the solution $\mathbf{x}^* = (\mathbf{I} - \mathbf{M})^{-1}\mathbf{b}$. Moreover, the residuals $\mathbf{F}(\mathbf{x}) = \mathbf{b} - (\mathbf{I} - \mathbf{M})\mathbf{x}$ converge to zero with a q-factor no larger than $c$.

*Proof.* In this proof the optimization problem is used in an important way. Given $\mathbf{x}_k$, we note that since $\sum \alpha_j^k = 1$, the new residual is

$$\begin{aligned}
\mathbf{F}(\mathbf{x}_{k+1}) &= b - (I - \mathbf{M})\mathbf{x}_{k+1} \\
&= \sum_{j=0}^{m_k} \alpha_j^k [b - (I - \mathbf{M})(b + \mathbf{M}\mathbf{x}_{k-m_k+j})] \\
&= \sum_{j=0}^{m_k} \alpha_j^k \mathbf{M}[b - (I - \mathbf{M})\mathbf{x}_{k-m_k+j}] \\
&= \mathbf{M} \sum_{j=0}^{m_k} \alpha_j^k \mathbf{F}(\mathbf{x}_{k-m_k+j}).
\end{aligned}$$

We take norms and use $\|\mathbf{M}\| = c$ to obtain

$$\|\mathbf{F}(\mathbf{x}_{k+1})\| \leq c \left\| \sum_{j=0}^{m_k} \alpha_j \mathbf{F}(\mathbf{x}_{k-m_k+j}) \right\|.$$

Optimality implies that

$$\left\| \sum_{j=0}^{m_k} \alpha_j \mathbf{F}(\mathbf{x}_{k-m_k+j}) \right\| \leq \|\mathbf{F}(\mathbf{x}_k)\|.$$

Hence

$$\|\mathbf{F}(\mathbf{x}_{k+1})\| \leq c\|\mathbf{F}(\mathbf{x}_k)\|, \tag{5.7}$$

as asserted. ☐

One might think that the analysis could proceed like that for Newton's method in that the result for the linear problem (convergence in one iteration for Newton) would imply a result for the nonlinear problem after a Taylor expansion if the initial iterate were accurate enough to neglect the high-order terms. In fact, that analogy is correct. We will illustrate the point with two theorems from Toth and Kelley (2015).

In the special case $m = 1$, one can solve the optimization problem analytically. One can use this to show that (5.7) holds for Anderson(1) in the nonlinear case if the initial iterate is sufficiently near the solution. The assumption of continuous differentiability is weaker than the one in Toth and Kelley (2015) and is also used in Chen and Kelley (2017).

### *5.2.2. The special case $m = 1$*

If $m = 1$ we need only assume that $\mathbf{G}$ is a continuously differentiable contraction to obtain q-linear convergence of the residuals.

**Assumption 5.3.** $\mathbf{G}$ has a fixed point $\mathbf{x}^*$.

- $\mathbf{G}$ is continuously differentiable in the ball $\mathcal{B}(\mathbf{x}^*, \hat{\rho}) = \{\mathbf{x} \mid \|\mathbf{e}\| \leq \hat{\rho}\}$ for some $\hat{\rho} > 0$.
- There is $c \in (0,1)$ such that for all $\mathbf{x}, \mathbf{y} \in \mathcal{B}(\mathbf{x}^*, \hat{\rho})$, $\|\mathbf{G}(\mathbf{x}) - \mathbf{G}(\mathbf{y})\| \leq c\|\mathbf{x} - \mathbf{y}\|$.

Theorem 5.4 is a generalization of a result from Toth and Kelley (2015) with stronger convergence and slightly weaker assumptions.

**Theorem 5.4.** Assume that Assumption 5.3 holds. Then if $\|e_0\|$ is sufficiently small, the Anderson(1) residuals with $\ell^2$-optimization converge q-linearly. Moreover,

$$\limsup_{k\to\infty} \frac{\|\mathbf{F}(\mathbf{x}_{k+1})\|}{\|\mathbf{F}(\mathbf{x}_k)\|} \leq c. \tag{5.8}$$

*Proof.* We will express the iteration as

$$\mathbf{x}_{k+1} = (1 - \alpha^k)\mathbf{G}(\mathbf{x}_k) + \alpha^k \mathbf{G}(\mathbf{x}_{k-1}), \tag{5.9}$$

and note that

$$\alpha^k = \frac{\mathbf{F}(\mathbf{x}_k)^T(\mathbf{F}(\mathbf{x}_k) - \mathbf{F}(\mathbf{x}_{k-1}))}{\|\mathbf{F}(\mathbf{x}_k) - \mathbf{F}(\mathbf{x}_{k-1})\|^2}. \tag{5.10}$$

Now define

$$\mathbf{a}_k = \mathbf{G}(\mathbf{x}_{k+1}) - \mathbf{G}((1 - \alpha^k)\mathbf{x}_k + \alpha^k \mathbf{x}_{k-1})$$

and

$$\mathbf{b}_k = \mathbf{G}((1 - \alpha^k)\mathbf{x}_k + \alpha^k \mathbf{x}_{k-1}) - \mathbf{x}_{k+1}.$$

Clearly

$$\mathbf{F}(\mathbf{x}_{k+1}) = \mathbf{G}(\mathbf{x}_{k+1}) - \mathbf{x}_{k+1} = \mathbf{a}_k + \mathbf{b}_k. \tag{5.11}$$

We will obtain an estimate of $\mathbf{F}(\mathbf{x}_{k+1})$ by estimating $\mathbf{a}_k$ and $\mathbf{b}_k$ separately.

By definition of the Anderson iteration (5.9) and contractivity of $G$,

$$\begin{aligned}\|\mathbf{a}_k\| &= \|\mathbf{G}(\mathbf{x}_{k+1}) - \mathbf{G}((1 - \alpha^k)\mathbf{x}_k + \alpha^k \mathbf{x}_{k-1})\| \\ &\leq c\|\mathbf{x}_{k+1} - (1 - \alpha^k)\mathbf{x}_k - \alpha^k \mathbf{x}_{k-1}\|\end{aligned}$$

$$
\begin{aligned}
&= c\|(1-\alpha^K)(\mathbf{G}(\mathbf{x}_K)-\mathbf{x}_K)-\alpha^K(\mathbf{G}(\mathbf{x}_{K-1})-\mathbf{x}_{K-1})\| \\
&= c\|(1-\alpha^k)\mathbf{F}(\mathbf{x}_k)-\alpha^k\mathbf{F}(\mathbf{x}_{k-1})\| \leq c\|\mathbf{F}(\mathbf{x}_k)\|,
\end{aligned} \tag{5.12}
$$

where the last inequality follows from the optimality property of the coefficients.

We now estimate $\mathbf{b}_k$. Now let

$$\delta_k = \mathbf{x}_{k-1} - \mathbf{x}_k.$$

To estimate $\mathbf{b}_k$ we note that

$$
\begin{aligned}
\mathbf{b}_k &= \mathbf{G}((1-\alpha^k)\mathbf{x}_k+\alpha^k\mathbf{x}_{k-1})-(1-\alpha^k)\mathbf{G}(\mathbf{x}_k)-\alpha^k\mathbf{G}(\mathbf{x}_{k-1}) \\
&= \mathbf{G}(\mathbf{x}_k+\alpha^k\delta_k)-\mathbf{G}(\mathbf{x}_k)+\alpha^k(\mathbf{G}(\mathbf{x}_k)-\mathbf{G}(\mathbf{x}_{k-1})) \\
&= \int_0^1 \mathbf{G}'(\mathbf{x}_k+t\alpha^k\delta_K)\alpha^k\delta_k\,\mathrm{d}t-\alpha^K\int_0^1 \mathbf{G}'(\mathbf{x}_k+t\delta_k)\delta_k\,\mathrm{d}t \\
&= \alpha^k\int_0^1 [\mathbf{G}'(\mathbf{x}_k+t\alpha^k\delta_k)-\mathbf{G}'(\mathbf{x}_k+t\delta_k)]\delta_k\,\mathrm{d}t.
\end{aligned} \tag{5.13}
$$

Since $\mathbf{G}'$ is continuous in $\mathcal{B}(\mathbf{x}^*,\rho)$, there is a non-decreasing function $\eta \in C[0,\infty)$ with $\eta(0)=0$ so that

$$\|\mathbf{G}'(\mathbf{x})-\mathbf{G}'(\mathbf{x}^*)\| \leq \eta(\|\mathbf{e}\|) \tag{5.14}$$

for all $\mathbf{x} \in \mathcal{B}(\mathbf{x}^*,\rho)$. Hence, if $\mathbf{x}_k$ and $\mathbf{x}_{k-1}$ are both in $\mathcal{B}(\mathbf{x}^*,\rho)$ (which is certainly true if $k=1$), then

$$\|\mathbf{b}_k\| \leq 2\eta(\|\mathbf{e}_k\|+\|\delta_k\|)\||\alpha^k|\|\delta_k\|. \tag{5.15}$$

Because $m=1$, it is not difficult to estimate $\alpha^k$. Clearly,

$$
\begin{aligned}
\mathbf{F}(\mathbf{x}_k)-\mathbf{F}(\mathbf{x}_{k-1}) = \mathbf{G}(\mathbf{x}_k)-\mathbf{G}(\mathbf{x}_{k-1})+\delta_k &= \delta_k-\int_0^1 \mathbf{G}'(\mathbf{x}_{k-1}-t\delta_k)\delta_k\,\mathrm{d}t \\
&= \left(I-\int_0^1 \mathbf{G}'(\mathbf{x}_{k-1}-t\delta_k)\,\mathrm{d}t\right)\delta_k.
\end{aligned}
$$

Since $\|\mathbf{G}'(u)\| \leq c$ for all $u \in \mathcal{B}(\mathbf{x}^*,\rho)$, we have

$$\|\delta_k\| \leq \|\mathbf{F}(\mathbf{x}_k)-\mathbf{F}(\mathbf{x}_{k-1})\|/(1-c). \tag{5.16}$$

Combine (5.16) and (5.10) to obtain

$$|\alpha^k|\|\delta_k\| \leq \frac{\|\mathbf{F}(\mathbf{x}_k)\|}{\|\mathbf{F}(\mathbf{x}_k)-\mathbf{F}(\mathbf{x}_{k-1})\|}\|\delta_k\| \leq \frac{\|\mathbf{F}(\mathbf{x}_k)\|}{1-c}. \tag{5.17}$$

Hence

$$\|\mathbf{F}(\mathbf{x}_{k+1})\| \leq \|\mathbf{F}(\mathbf{x}_k)\|(c+2\eta(\|\mathbf{e}_k\|+\|\delta_k\|)\|). \tag{5.18}$$

So, given $c < \hat{c} < 1$, we may reduce $\rho$ if needed so that

$$\|\mathbf{F}(\mathbf{x}_{k+1})\| \leq \hat{c}\|\mathbf{F}(\mathbf{x}_k)\|. \tag{5.19}$$

We complete the proof by combining (5.18) and (5.19) to show that

$$\|\mathbf{F}(\mathbf{x}_{k+1})\| \leq \|\mathbf{F}(\mathbf{x}_k)\|(c+o(1))$$

as $k \to \infty$. □

### *5.2.3. Local convergence for general m*

We must either assume or arrange that the $\ell^1$-norm of the coefficients be uniformly bounded to prove local convergence for $m > 1$ or for any norm other than the $\ell^2$-norm. In addition, the convergence is r-linear rather than q-linear. We will state Chen and Kelley's (2017) extension of the local convergence result from Toth and Kelley (2015). We refer to those papers for the proof.

**Theorem 5.5.** Assume that Assumption 5.3 holds. Assume that there is $M_\alpha$ such that, for all $k \geq 0$,

$$\sum_{j=1}^{m_k} |\alpha_j| \leq M_\alpha. \tag{5.20}$$

Then if $\mathbf{x}_0$ is sufficiently near to $\mathbf{x}^*$, the Anderson iterations converge, and

$$\limsup_{k\to\infty} \left(\frac{\|\mathbf{F}(\mathbf{x}_k)\|}{\|\mathbf{F}(\mathbf{x}_0)\|}\right)^{1/k} \leq c. \tag{5.21}$$

The assumption (5.20) that the $\ell^1$-norm of the coefficients is bounded can be enforced within the iteration by controlling $m_k$. One way to do this is to reduce $m_k$ if the $\ell^1$-norm of the coefficients or (in the $\ell^2$-norm case) the conditioning of the least-squares problem (5.6) exceeds a predetermined bound. Walker and Ni (2011) and An *et al.* (2017) advocate limiting the condition number of the least-squares problem (5.6). Toth (2016) shows that if one does this the r-linear convergence improves to q-linear. However, the performance in practice of methods that enforce (5.20) is not always better than the original version. In Section 5.3 we discuss another approach which requires that the coefficients be non-negative (so $M_\alpha \equiv 1$) as a way to globalize convergence.

### *5.2.4. H-equation example*

We return to the $H$-equation with some of the results from Toth and Kelley (2015). This computation exposes one of the many mysteries in Anderson acceleration.

We begin with a look at Newton–GMRES and Picard iteration, which are well known to converge even in the singular $c = 1$ case. If $c = 1$ the fixed-point map is not a contraction and, as we have seen, the Jacobian is singular at the solution. However, if the initial iterate is well chosen ($H_0(\mu) \equiv 1$ is a good choice), both the Picard and Newton iterations will converge.

Table 5.1. Function evaluations for Newton–GMRES and fixed-point iteration.

|  | Newton–GMRES |  |  | Fixed point |  |  |
|---|---|---|---|---|---|---|
| $\omega$ | 0.5 | 0.99 | 1.0 | 0.5 | 0.99 | 1.0 |
| $F$s | 12 | 18 | 49 | 11 | 75 | 23 970 |

Table 5.2. Iteration statistics for Anderson($m$).

|  | $m = 1$ |  |  | $m = 2$ |  |  | $m = 5$ |  |  |
|---|---|---|---|---|---|---|---|---|---|
| $\omega$ | ITS | $\kappa$ | $S_{\max}$ | ITS | $\kappa$ | $S_{\max}$ | ITS | $\kappa$ | $S_{\max}$ |
| 0.50 | 7 | $1.0\times 10^{0}$ | 1.4 | 6 | $2.9\times 10^{3}$ | 1.4 | 6 | $2.5\times 10^{10}$ | 1.4 |
| 0.99 | 11 | $1.0\times 10^{0}$ | 4.0 | 10 | $9.8\times 10^{3}$ | 5.4 | 12 | $1.6\times 10^{11}$ | 5.4 |
| 1.00 | 21 | $1.0\times 10^{0}$ | 3.0 | 16 | $2.9\times 10^{3}$ | 14.3 | 27 | $8.0\times 10^{9}$ | 14.8 |

The computations in this section are from an $N = 500$ point mesh. We terminated the iterations when $\|\mathbf{F}(\mathbf{h}_n)\| \leq 10^{-8}\|\mathbf{F}(\mathbf{h}_0)\|$. Table 5.1 reports cost in terms of function evaluations for both Newton–GMRES and Picard iterations for three values if $\omega$. One can see the effect of the singular Jacobian for both Newton–GMRES and Picard iteration. In the statistics for Newton–GMRES we count both the function evaluations in the nonlinear iterations and those used in a finite-difference Jacobian-vector product.

In Table 5.2 we report the cost of Anderson($m$) with $m = 1, 2, 5$ for the same problems. We tabulate the number of iterations ITS needed to terminate, the maximum condition number $\kappa$ of the least-squares problem, and the maximum $\ell^1$-norm $S_{\max}$ of the coefficients. $S_{\max}$ is not large and, at least for this problem, the assumption that (5.20) holds is reasonable.

Anderson($m$) does far better for these problems than the theory predicts, costing less than even Newton–GMRES. Note that each iteration of Anderson costs a single function evaluation. For $m = 5$ the least-squares problems become very ill-conditioned, which is unsurprising given that $\mathbf{G}'$ is an integral operator. This ill-conditioning does not cause the iteration to fail, but does, especially when $\omega = 1$, increase the cost.

In Figure 5.1 we plot the residual histories for Anderson(1) and the three values of $\omega$. The reader should compare Figure 5.1 to Figure 2.5.

Toth *et al.* (2017) contains mesh independence results for Anderson acceleration similar to those in Section 3.3.

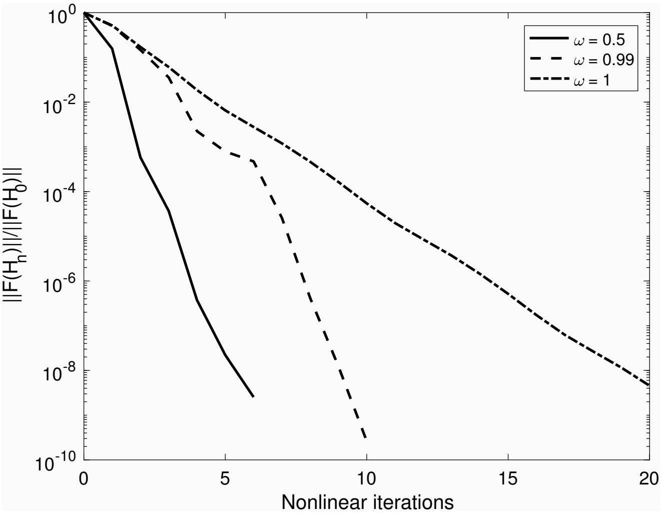

Figure 5.1. Anderson(1) for the $H$-equation example.

### 5.3. *The EDIIS globalization*

Anderson acceleration performs poorly for some applications. One example is electronic structure computations for metallic systems where the HOMO–LUMO gap is small (Kudin, Scuseria and Cancès 2002). In such cases one must use a small mixing parameter to ensure convergence. However, a small mixing parameter degrades the performance of the iteration. In addition, an accurate initial iterate is often necessary, and finding an acceptable initial iterate is often problematic.

One attempt to solve these problems for small systems is the EDIIS (energy DIIS) algorithm from Kudin *et al.* (2002). This is the form of Anderson acceleration in the Gaussian computational chemistry code (Frisch *et al.* 2009). EDIIS differs from Anderson acceleration in that the fixed-point problem is designed differently to minimize energy and a non-negativity constraint on the coefficients. So, the optimization problem becomes

$$\min \left\| \sum_{j=0}^{m_k} \alpha_j^k \mathbf{F}_{k-m_k+j} \right\| \quad \text{subject to } \sum_{j=0}^{m_k} \alpha_j^k = 1, \ \alpha_j^k \geq 0. \tag{5.22}$$

The optimization problem (5.22) for EDIIS is much harder than the linear least-squares problem (5.6) in algorithm **anderson**. The approach of Kudin *et al.* (2002) is a direct examination of the boundary of the feasible simplex, which is not practical for a depth much greater than $m = 3$. Since $m$ is small in practice, expressing the optimization problem as a bound-constrained quadratic program is an efficient alternative. Moré and Toraldo (1991) survey the literature on this topic. For example, a bound-constrained quadratic programming code such as the MINQ code (Neumaier 1998) is a reasonable choice. However, this approach squares the condition number. The classic method from Golub and Saunders (1969) uses an active set method and QR factorization to avoid this.

In Chen and Kelley (2017) we prove that adding the non-negativity constraint condition to the optimization makes the algorithm globally convergent. Theorem 5.6 and its proof make that precise.

**Theorem 5.6.** Let $\mathbf{G}$ be a contraction on a convex $D \subset \mathbb{R}^N$ with contractivity constant $c$. Let $\mathbf{x}^*$ be the unique fixed point of $\mathbf{g}$ in $D$. Then, for any $\mathbf{x}_0 \in D$, EDIIS($m$) converges to $\mathbf{x}^*$ r-linearly with r-factor

$$\hat{c} = c^{1/(m+1)}.$$

In fact,

$$\|\mathbf{e}_k\| \leq \hat{c}^k \|\mathbf{e}_0\|. \tag{5.23}$$

*Proof.* The proof does not use the optimality condition and only requires that the iteration $\{\mathbf{x}_k\}$ have the form

$$\mathbf{x}_{k+1} = \sum_{j=0}^{m_k} \alpha_j^k \mathbf{G}(\mathbf{x}_{k-m_k+j}), \tag{5.24}$$

where $m_k \leq m$, $\alpha_j^k \geq 0$, and $\sum_{j=0}^{m_k} \alpha_j^k = 1$.

We induct on $k$. Clearly (5.23) holds for both $m_k = 0$, by definition, and $k = m_k = 1$ because the iteration in that case is a single Picard iteration. Assume that the result holds for $k \leq K$. Then (5.24) and $\sum_{j=0}^{m_k} \alpha_j^k = 1$ imply that

$$\mathbf{e}_{K+1} = \sum_{j=0}^{m_k} \alpha_j^k (\mathbf{G}(\mathbf{x}_{k-m_k+j}) - \mathbf{x}^*).$$

Non-negativity of the $\alpha^k$ then implies that

$$\begin{aligned} \|\mathbf{e}_{K+1}\| &\leq \sum_{j=0}^{m_K} \alpha_j^K \|\mathbf{G}(\mathbf{x}_{K-m_K+j}) - \mathbf{x}^*\| \\ &\leq \sum_{j=0}^{m_K} \alpha_j^K c \|\mathbf{x}_{K-m_K+j} - \mathbf{x}^*\| \\ &\leq c \sum_{j=0}^{m_K} \alpha_j^K \hat{c}^{K-m_K+j} \|\mathbf{e}_0\| \leq \hat{c}^{K+1}(c\hat{c}^{-m-1}) \leq \hat{c}^{K+1}. \end{aligned}$$ ☐

The theorem says the iteration history of $m$ vectors will eventually be arbitrarily close to $\mathbf{x}^*$. Hence restarting the iteration after sufficiently many EDIIS iterations will result in local convergence at the rate predicted by

Theorem 5.5, which is better than (5.23). However, it is not clear how to decide when to restart. Theorem 5.7, the local convergence result from Chen and Kelley (2017), says that one can simply continue with the EDIIS iteration and the local convergence estimate will hold.

**Theorem 5.7.** Let the assumptions of Theorem 5.6 hold. Then the EDIIS algorithm converges to $\mathbf{x}^*$ and (5.21) holds.

The proof of Theorem 5.7 depends strongly on Theorem 5.6 to generate good initial data and the constrained optimization problem to guarantee that the $\ell^1$-norm of the coefficients is bounded.

## 6. Newton's method for semi-smooth functions

The results in this section generalize Newton's method in a way that does not require differentiability of $\mathbf{F}$. Many of these results extend to a function space setting, but only with some significant modifications in the analysis. Several papers, for example those of Chen, Nashed and Qi (2001), Hintermüller and Ulbrich (2003) and Ulbrich (2011), generalize these ideas to function spaces.

### *6.1. Generalized derivatives and semi-smooth functions*

We begin with a idealized version of Newton's method for Lipschitz continuous functions. Let $\Omega \subset \mathbb{R}^N$ be open. Suppose $\mathbf{F} : \Omega \to \mathbb{R}^N$ is locally Lipschitz continuous. Rademacher's theorem says that $\mathbf{F}$ is Fréchet differentiable almost everywhere. The proof of this remarkable result can be found in Federer (1969, Theorem 3.1.6).

We let $D_{\mathbf{F}}$ denote the set of points where $\mathbf{F}$ is Fréchet differentiable. The *generalized Jacobian* (Clarke 1990) of $\mathbf{F}$ at $\mathbf{u} \in \mathbb{R}^N$ is the set

$$\partial \mathbf{F}(\mathbf{u}) = \mathrm{co}\Big\{ \lim_{\mathbf{u}_j \to \mathbf{u}; \mathbf{u}_j \in D_{\mathbf{F}}} \mathbf{F}'(\mathbf{u}_j) \Big\}, \tag{6.1}$$

where co denotes the closed convex hull.

Consider the scalar function $f(x) = |x|$. The function is differentiable except at $x = 0$. At $x = 0$, where $f$ is not differentiable, the generalized derivative is the interval $[-1, 1]$.

The chain rule is not a trivial matter for non-smooth problems, and there are several variations. As an example we will state Theorem 2.6.6 from Clarke (1990) and one of its corollaries. We state the results using the formulation from Hintermüller (2010).

In the theorem $\mathbf{F} \circ \mathbf{G}$ will denote the composition $\mathbf{F} \circ \mathbf{G}(\mathbf{x}) \equiv \mathbf{F}(\mathbf{G}(\mathbf{x}))$.

**Theorem 6.1.** Let $\mathbf{Q} = \mathbf{G} \circ \mathbf{F}$, where $\mathbf{F} : \mathbb{R}^N \to \mathbb{R}^N$ is Lipschitz continuous in a neighbourhood of $\mathbf{x}$ and $\mathbf{G} : \mathbb{R}^N \to \mathbb{R}^P$ is Lipschitz continuous in a neighbourhood of $\mathbf{F}(\mathbf{x})$. Then $\mathbf{Q}$ is Lipschitz continuous in a neighbourhood of $\mathbf{x}$ and, for all $\mathbf{v} \in \mathbb{R}^N$,

$$\partial \mathbf{Q}(\mathbf{x})\mathbf{v} \subset \mathrm{co}\{\partial \mathbf{G}(\mathbf{F}(\mathbf{x}))\partial \mathbf{F}(\mathbf{x})\mathbf{v}\}. \tag{6.2}$$

Moreover, if $\mathbf{G}$ is continuously differentiable, then the inclusion is an equality, that is, for all $\mathbf{v} \in \mathbb{R}^N$,

$$\partial \mathbf{Q}(\mathbf{x})\mathbf{v} = \mathbf{G}'(\mathbf{F}(\mathbf{x}))\partial \mathbf{F}(\mathbf{x})\mathbf{v}. \tag{6.3}$$

If $\mathbf{G}$ is real-valued ($P = 1$), then the vector $\mathbf{v}$ can be omitted from (6.2) and (6.3).

One needs more than Lipschitz continuity to properly generalize convergence theorems for Newton's method. Mifflin (1977) introduced the concept of semi-smoothness in the context of optimization. The extension to nonlinear equations in Qi and Sun (1993) was the beginning of a very lively research area.

We will use one of the several equivalent definitions from Qi and Sun (1993).

**Definition 6.2.** $\mathbf{F}$ is semi-smooth at $\mathbf{x} \in \mathbb{R}^N$ if $\mathbf{F}$ is locally Lipschitz continuous and, for all $\mathbf{w} \in \mathbb{R}^N$ and $\mathbf{V} \in \partial \mathbf{F}(\mathbf{x}+\mathbf{w})$,

$$\mathbf{F}(\mathbf{x}+\mathbf{w}) - \mathbf{F}(\mathbf{x}) - \mathbf{V}\mathbf{w} = o(\|\mathbf{w}\|) \tag{6.4}$$

as $\mathbf{w} \to 0$. $\mathbf{F}$ is semi-smooth of order $p$ if

$$\mathbf{F}(\mathbf{x}+\mathbf{w}) - \mathbf{F}(\mathbf{x}) - \mathbf{V}\mathbf{w} = O(\|\mathbf{w}\|^{1+p}) \tag{6.5}$$

as $\mathbf{w} \to 0$.

There is a subtle and important point in the definition. The operator $\mathbf{V}$ is in $\partial \mathbf{F}(\mathbf{x}+\mathbf{w})$, not, as one might expect, in $\partial \mathbf{F}(\mathbf{x})$. This point is the critical difference between semi-smoothness and differentiability.

We will make use of a few facts from Mifflin (1977), Qi and Sun (1993) and Clarke (1990).

- If $\mathbf{F}$ is semi-smooth at $\mathbf{x}$, then the directional derivatives $\mathrm{d}\mathbf{F}(\mathbf{x}:\mathbf{u})$ (2.29) exist for all directions $\mathbf{u}$.
- The composition of two semi-smooth functions is semi-smooth.

### *6.2. Local convergence of Newton's method*

We use Newton's method via

$$\mathbf{x}_+ = \mathbf{x}_c - \mathbf{V}_c^{-1}\mathbf{F}(\mathbf{x}_c), \tag{6.6}$$

where $\mathbf{V}_c$ is any member of $\partial \mathbf{F}(\mathbf{x}_c)$. We will state the results in terms of an inexact formulation,

$$\mathbf{x}_+ = \mathbf{x}_c + \mathbf{s}, \tag{6.7}$$

where

$$\|\mathbf{V}_c\mathbf{s} + \mathbf{F}(\mathbf{x}_c)\| \le \eta_c \|\mathbf{F}(\mathbf{x}_c)\| \tag{6.8}$$

and $\mathbf{V}_c \in \partial\mathbf{F}(\mathbf{x}_c)$. This iteration does not converge, even locally, for general Lipschitz functions. The iteration does converge for semi-smooth functions. We will state a convergence theorem which combines results from Qi and Sun (1993), Pang and Qi (1993), Martinez and Qi (1995) and Facchinei, Fischer and Kanzow (1996), and extends Theorem 2.4 to the semi-smooth case.

We must formulate an analogue to the standard assumptions and then use that to argue that the Newton sequence exists if the initial iterate is sufficiently near a solution $\mathbf{x}^*$.

**Assumption 6.3.** There is $\mathbf{x}^* \in \mathbb{R}^N$ and $\rho^* > 0$ such that

- $\mathbf{F}(\mathbf{x}*) = 0$;
- $\mathbf{F}$ is semi-smooth in $\mathcal{B}(\mathbf{x}^*, \rho^*)$;
- every element of $\partial\mathbf{F}(\mathbf{x}^*)$ is non-singular.

In the classical case one uses Lipschitz continuity of $\mathbf{F}'$ to argue that $\mathbf{F}'(\mathbf{x})$ is non-singular for all $\mathbf{x}$ sufficiently near $\mathbf{x}^*$ and then to prove quadratic convergence. In fact, only local Lipschitz continuity is needed for the first assertion (Qi and Sun 1993). We state this fact formally as Lemma 6.4.

**Lemma 6.4.** Let $\mathbf{F}$ be Lipschitz continuous in a neighbourhood of $\mathbf{x}$ and let all matrices in $\partial\mathbf{F}(\mathbf{x})$ be non-singular. Then there are $\rho$ and $C > 0$ such that for all $\mathbf{y} \in \mathcal{B}(\mathbf{x}, \rho)$ and all $\mathbf{V} \in \partial\mathbf{F}(\mathbf{y})$

$$\|\mathbf{V}^{-1}\| \le C. \tag{6.9}$$

**Theorem 6.5.** Let $\mathbf{F} : \mathbb{R}^N \to \mathbb{R}^N$ with $\mathbf{F}(\mathbf{x}^*) = 0$. Assume that $\mathbf{F}$ is semi-smooth at $\mathbf{x}^*$ and that all matrices in $\partial\mathbf{F}(\mathbf{x}^*)$ are non-singular. Then there are $\bar\eta, \bar\delta, K > 0$ such that if $\mathbf{x}_0 \in \mathcal{B}(\mathbf{x}^*, \bar\delta)$ and $\eta_n \le \bar\eta$, then the generalized inexact Newton iteration (6.7) converges to $\mathbf{x}^*$ and

$$\|\mathbf{e}_+\| \le K\eta_c\|\mathbf{e}_c\| + o(\|\mathbf{e}_c\|).$$

Moreover, if $\mathbf{F}$ is semi-smooth of order $0 < p \le 1$ at $x^*$, then

$$\|\mathbf{e}_+\| \le K(\eta_c\|\mathbf{e}_c\| + \|\mathbf{e}_c\|^{1+p}).$$

*Proof.* We will prove the special case where $\eta_n \equiv 0$ and $\mathbf{F}$ is semi-smooth of order 1. We refer to Qi and Sun (1993), Pang and Qi (1993), Martinez and Qi (1995) and Facchinei *et al.* (1996) for the complete analysis. We will follow the formulation in Hintermüller (2010).

Let $\alpha \in (0, 1)$ be arbitrary. We will proceed as we did in the proof of the classical theorem (Theorem 2.3) by first showing that the error converges to zero q-linearly with q-factor $\le \alpha$. We will then extract the quadratic convergence from semi-smoothness of order 1. Let $\mathbf{x}_c$ be near enough to $\mathbf{x}^*$ so that (6.9) holds. There is $\mathbf{V}_c \in \partial\mathbf{F}(\mathbf{x}_c)$ such that

$$\mathbf{e}_+ = \mathbf{e}_c - \mathbf{V}_c^{-1}\mathbf{F}(\mathbf{x}_c) = \mathbf{V}_c^{-1}(\mathbf{V}_c\mathbf{e}_c - (\mathbf{F}(\mathbf{x}_c) - \mathbf{F}(\mathbf{x}^*))). \tag{6.10}$$

Semi-smoothness of order 1 implies that

$$\mathbf{V}_c\mathbf{e}_c - (\mathbf{F}(\mathbf{x}_c) - \mathbf{F}(\mathbf{x}^*)) = O(\|\mathbf{e}_c\|^2) \tag{6.11}$$

for $\mathbf{x}_c$ sufficiently near $\mathbf{x}^*$. In particular, we may require $\mathbf{x}_c$ to be near enough to $\mathbf{x}^*$ so that

$$\|\mathbf{V}_c\mathbf{e}_c - (\mathbf{F}(\mathbf{x}_c) - \mathbf{F}(\mathbf{x}^*))\| \leq \frac{\alpha}{C}\|\mathbf{e}_c\|,$$

where $C$ is the bound on $\|\mathbf{V}^{-1}\|$ from (6.9). Hence (6.10) implies that $\|\mathbf{e}_+\| \leq \alpha\|\mathbf{e}_c\|$, proving convergence. Quadratic convergence then follows from (6.11). □

Qi and Sun (1993) also prove a generalization of the Kantorovich theorem for semi-smooth functions. The reader should compare this theorem to Theorem 3.2.

**Theorem 6.6.** Let $\mathbf{F}$ be locally Lipschitz and semi-smooth on $\mathcal{B}(\mathbf{x}_0, r)$. Suppose there are $\beta, \gamma, \delta > 0$ such that for any $\mathbf{V} \in \partial\mathbf{F}(\mathbf{x})$ and $\mathbf{x}, \mathbf{y} \in \mathcal{B}(\mathbf{x}_0, r)$ we have:

- $\mathbf{V}$ is non-singular and $\|\mathbf{V}^{-1}\| \leq \beta$;
- $\|\mathbf{V}(\mathbf{x} - \mathbf{y}) - \mathrm{d}\mathbf{F}(\mathbf{x} : \mathbf{x} - \mathbf{y})\| \leq \gamma\|\mathbf{y} - \mathbf{x}\|$;
- $\|\mathbf{F}(\mathbf{x}) - \mathbf{V}(\mathbf{y}) - \mathrm{d}\mathbf{F}(\mathbf{x} : \mathbf{x} - \mathbf{y})\| \leq \delta\gamma\|\mathbf{y} - \mathbf{x}\|$;
- $\alpha = \beta(\gamma + \delta) < 1$, and
- $\beta\|\mathbf{F}(\mathbf{x}_0)\| \leq r(1 - \alpha)$.

Then the semi-smooth Newton iteration

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \mathbf{V}_n^{-1}\mathbf{F}(\mathbf{x}_n),$$

with $\mathbf{V}_n \in \partial\mathbf{F}(\mathbf{x}_n)$, remains in $\mathcal{B}(\mathbf{x}_0, r)$, converges to a solution $\mathbf{x}^*$, and

$$\|\mathbf{x}_n - \mathbf{x}^*\| \leq \frac{\alpha}{1 - \alpha}\|\mathbf{x}_n - \mathbf{x}_{n-1}\|.$$

### *6.3. Global convergence*

Hintermüller (2010), motivated by problems in optimal control, presents examples for which semi-smooth Newton converges from any starting point. Those examples are a special case and in general one cannot expect the initial iterate to be accurate enough for the local convergence theory to hold. While there is no globalization method that applies to all semi-smooth problems, there are some easy-to-implement approaches which can be used for many classes of problems. We describe some such methods in Sections 6.3.1, 6.4, and 7.1.

Trust region methods have been applied to semi-smooth problems in function spaces (Heinkenschloss, Ulbrich and Ulbrich 1999, Ulbrich 2001), especially those connected to constrained optimal control problems for partial differential equations.

### *6.3.1. Line search methods for complementarity problems*

For smooth problems the line search methods from Section 2.7 are by far the most common solution to the problem of a poor initial iterate. Line search methods have only succeeded for limited classes of semi-smooth equations (De Luca, Facchinei and Kanzow 1996, Jiang and Qi 1997). Nonlinear complementarity problems are an example of such problems. A nonlinear complementarity problem is to find $\mathbf{x}$ such that

$$\mathbf{x} \geq 0, \quad \mathbf{P}(\mathbf{x}) \geq 0, \quad \mathbf{x}^T\mathbf{P}(\mathbf{x}) = 0. \tag{6.12}$$

In (6.12) the inequalities are componentwise and $\mathbf{P}$ is continuously differentiable. The approach is to transform (6.12) into a semi-smooth nonlinear equation.

The method in De Luca *et al.* (1996) uses the Fischer–Burmeister function

$$\phi(t, s) = \sqrt{t^2 + s^2} - (t + s) \tag{6.13}$$

(Fischer 1992). It is easy to show that $\phi(t, s) = 0$ if and only if $t \geq 0$, $s \geq 0$ and $st = 0$. We extend the definition $\phi$ from $R^2$ to $\mathbb{R}^N \times \mathbb{R}^N$ via componentwise application to obtain

$$\Phi(\mathbf{x}, \mathbf{y}) \equiv \begin{pmatrix} \phi(x_1, y_1) \\ \phi(x_2, y_2) \\ \vdots \\ \phi(x_N, y_N) \end{pmatrix}.$$

Then it is easy to verify that (6.12) is equivalent to the semi-smooth equation

$$\mathbf{F}(\mathbf{x}) \equiv \Phi(\mathbf{x}, \mathbf{P}(\mathbf{x})) = 0. \tag{6.14}$$

We compute $\partial\mathbf{F}(\mathbf{x})$ to show how one does this in cases where the componentwise application of a semi-smooth scalar function is composed with a smooth function. De Luca *et al.* (1996) have a very nice description. The goal is to compute $\mathbf{J} \in \partial\mathbf{F}(\mathbf{x})$. Any $\mathbf{J}$ computed with algorithm **compute_J** will be in $\partial\mathbf{F}(\mathbf{x})$.

Here, as in Section 2, $\mathbf{u}_i$ is the unit vector in the $i$th coordinate direction.

De Luca *et al.* (1996) propose an algorithm that differs from algorithm **newton_armijo** in only a few ways. The descent direction $\mathbf{d}$ is the solution of

$$\mathbf{J}\mathbf{d} = -\mathbf{F}(\mathbf{x}),$$

```
compute_J
    Set $\mathcal{A} = \{i \mid x_i = 0 = p_i(\mathbf{x})\}$.
    Let $\mathbf{z} \in \mathbb{R}^N$ be such that $z_i \neq 0$ for all $i \in \mathcal{A}$.
    For $i \notin \mathcal{A}$ set the $i$th row of $\mathbf{J}$ to be
        $\left(\frac{x_i}{\sqrt{x_i^2 + p_i(\mathbf{x})^2}} - 1\right)\mathbf{u}_i + \left(\frac{p_i(\mathbf{x})}{\sqrt{x_i^2 + p_i(\mathbf{x})^2}} - 1\right)\nabla p_i(\mathbf{x}).$
    For $i \in \mathcal{A}$ set the $i$th row of $\mathbf{J}$ to be
        $\left(\frac{z_i}{\sqrt{z_i^2 + (\nabla p_i(\mathbf{x})^T z)^2}} - 1\right)\mathbf{u}_i + \left(\frac{\nabla p_i(\mathbf{x}) z_i}{\sqrt{z_i^2 + (\nabla p_i(\mathbf{x})^T z)^2}} - 1\right)\nabla p_i(\mathbf{x}).$
```

where $\mathbf{J}$ is any element of $\partial\mathbf{F}(\mathbf{x})$. In algorithm **newton_armijo** we asked for sufficient decrease of $\|\mathbf{F}\|$. The analysis is more subtle in the non-smooth case because $\|\mathbf{F}\|$ is not differentiable. The algorithm in De Luca *et al.* (1996) resolves this problem by observing that

$$\Psi(\mathbf{x}) = \|\mathbf{F}(\mathbf{x})\|_2^2$$

is smooth and $\mathbf{d}$ is a descent direction for $\Psi$ because

$$\nabla\Psi(\mathbf{x})\mathbf{d} = -2\|\mathbf{F}(\mathbf{x})\|_2^2.$$

Now one can proceed as in the smooth case. The smoothness of $\Psi$ implies that $\mathbf{d}$ is a descent direction for $\|\mathbf{F}\|$ and is key to the success of the line search. All of these properties depend on the special structure of the nonlinear complementarity problem and the Fischer–Burmeister function. The convergence theory is subtle and more complicated than Theorem 2.6.

The algorithm in De Luca *et al.* (1996) includes tests for singularity or ill-conditioning of $\mathbf{J}$, thereby explicitly avoiding one of the failure modes in Theorem 2.6. One result, which is very like Theorem 2.6, is that if the sequence of iterations is bounded, one limit point of that sequence $\mathbf{x}^*$ is a solution of $\mathbf{F}(\mathbf{x}) = 0$, and $\mathbf{P}$ is Lipschitz continuously differentiable, then $\mathbf{x}_n \to \mathbf{x}^*$ q-quadratically.

### *6.4. Smoothing function methods*

Another more general way to globalize the semi-smooth Newton iteration is the smoothing function approach. The idea was developed in the context of variational inequalities (Chen, Qi and Sun 1998, Chen and Ye 1999) and generalized to more general nonlinearities and infinite dimensions in Chen *et al.* (2001). Here we approximate $\mathbf{F}(\mathbf{x})$ by a family of functions $\tilde{\mathbf{F}}(\mathbf{x}, \epsilon)$, where $\tilde{\mathbf{F}}$ is a Lipschitz continuously differentiable function of $\mathbf{x}$ for $\epsilon > 0$ and $\tilde{\mathbf{F}}(\mathbf{x}, 0) = \tilde{\mathbf{F}}(\mathbf{x})$. We require that $\tilde{\mathbf{F}}$ satisfy the *smoothing approximation* *property*

$$\|\tilde{\mathbf{F}}(\mathbf{x},\epsilon) - \mathbf{F}(\mathbf{x})\| \le \mu\epsilon \tag{6.15}$$

for some $\mu > 0$.

For example, one can smooth the Fischer–Burmeister function (6.13) with

$$\phi(t,s,\epsilon) = \sqrt{t^2 + s^2 + \epsilon^2} - (t+s).$$

Chen (2000) discusses several ways to apply smoothing methods to complementarity problems.

The iteration is

$$\mathbf{x}_+ = \mathbf{x}_c + \lambda_c \mathbf{d}_c, \tag{6.16}$$

where

$$\mathbf{d}_c = -\tilde{\mathbf{F}}'(\mathbf{x}_c,\epsilon_c)^{-1}\mathbf{F}(\mathbf{x}). \tag{6.17}$$

In (6.16) $\tilde{\mathbf{F}}'$ is the Jacobian in the $\mathbf{x}$ variables. The stepsize $\lambda$ satisfies an interesting hybrid sufficient decrease condition

$$\|\tilde{\mathbf{F}}(\mathbf{x}_c + \lambda\mathbf{d}_c, \epsilon_c)\|^2 \le \|\tilde{\mathbf{F}}(\mathbf{x}_c,\epsilon_c)\|^2 - \alpha\lambda\|\mathbf{F}(\mathbf{x}_c)\|^2.$$

The algorithms have several ways to update $\epsilon$. In Chen *et al.* (1998) the update can be one of $\epsilon_+ = O(\|\mathbf{F}(\mathbf{x}_+)\|)$, $\epsilon_+ = \epsilon_c/2$ or $\epsilon_+ = \epsilon_c$, depending on the rate of decrease in $\mathbf{F}(\mathbf{x})$.

One can obtain superlinear convergence if the approximations satisfy the *Jacobian consistency* property,

$$\lim_{\epsilon\to 0} \inf_{\mathbf{V}\in\partial\mathbf{F}(\mathbf{x})} \|\tilde{\mathbf{F}}'(\mathbf{x},\epsilon) - \mathbf{V}\| = 0 \tag{6.18}$$

for all $\mathbf{x}$.

Smoothing methods are, at least for local convergence, related to the splitting methods from Chen and Yamamoto (1989) and their measure-theoretic extensions in Heinkenschloß, Kelley and Tran (1992), Kelley (1994) and Kelley and Sachs (1994). These methods apply to problems for which the generalized Jacobian can be well approximated by the Jacobian $\mathbf{J}$ of a nearby smooth map. The iteration is

$$\mathbf{x}_+ = \mathbf{x}_c - \mathbf{J}(\mathbf{x}_c)^{-1}\mathbf{F}(\mathbf{x}).$$

Coffey, McMullan, Kelley and McRae (2003*b*) consider one example of such a problem where $\mathbf{F}$ is a second-order approximation to the Euler equations. The non-smoothness arises from a flux limiter. The map $\mathbf{J}$ is the Jacobian for a smooth first-order approximation of the same problem. Coffey *et al.* (2003*b*) globalized the iteration with pseudo-transient continuation.

Smoothing methods have also been globalized with trust region methods. Yang and Qi (2005) approach the nonlinear complementarity problem in this way. In Section 7.2 we will globalize a semi-smooth equation with a different kind of continuation.

## 7. Pseudo-transient continuation

Pseudo-transient continuation ($\Psi$TC) is an algorithm for finding stable steady-state solutions of time-dependent equations, such as

$$\dot{\mathbf{x}} = -\mathbf{F}(\mathbf{x}). \tag{7.1}$$

In (7.1) $\dot{\mathbf{x}} = \mathrm{d}\mathbf{x}/\mathrm{d}t$ and the minus sign before $\mathbf{F}$ is a convention. A steady-state solution $\mathbf{x}^*$ is time-independent, so $\dot{\mathbf{x}}^* = 0$. The solution is stable if the solution of the initial value problem for (7.1) with initial data sufficiently near $\mathbf{x}^*$ converges to $\mathbf{x}^*$ as $t \to \infty$. We will consider only linear stability and ask that the eigenvalues of $\mathbf{F}'(\mathbf{x}^*)$ be positive. One might think that one could simply apply Newton's method to the nonlinear equation $\mathbf{F}(\mathbf{x}) = 0$ and solve the problem, but that would be wrong. The reason is that not all solutions of $\mathbf{F}(\mathbf{x}) = 0$ are dynamically stable.

A simple example will illustrate the ideas. Consider the parameter-dependent scalar equation

$$\dot{x} = -(x^3 - \lambda x). \tag{7.2}$$

When $\lambda \leq 0$, the function $x \equiv 0$ is the only steady-state solution and it is stable since $f'(0) = -\lambda \geq 0$. When $\lambda > 0$, however, there are three steady-state solutions,

$$x \equiv \pm\sqrt{\lambda} \quad \text{and} \quad x \equiv 0.$$

The two non-zero solutions are stable and $x \equiv 0$ is not. If one solves $f(x) = 0$ with Newton's method, the iteration is

$$x_+ = \frac{-2x_c^3}{\lambda - 3x_c^2}.$$

Hence Newton's method will converge to the unstable solution if the initial iterate $x_0$ is sufficiently small.

The solution of the initial value problem, on the other hand, will converge to one of the stable steady-state solutions if $x(0) \neq 0$. One way to find the stable steady-state solution would be numerical integration with Euler's method:

$$x_{k+1} = x_k - hf(x_k) = x_k - h(x_k^3 - \lambda x_k) = (1 + h\lambda)x_k - hx_k^3.$$

It is easy to see that $x_{k+1} > x_k$ if $x_k > 0$ is small. Hence the numerical integration converges to the stable steady-state solution for sufficiently small $h$. While this would succeed in finding a stable steady state, the cost would be an accurate simulation in time, which may not be of interest if only the steady-state solution is needed.

$\Psi$TC is a way to move from the time-accurate simulation to a Newton iteration by managing a pseudo-timestep, which one can think of as a continuation parameter (or a trust region parameter: Higham 1996). The method updates the timestep as the iteration progresses with the objective of making the iteration converge superlinearly near the solution. ΨTC has been applied in aerodynamics (Venkatakrishnan 1989), hydrology (Farthing *et al.* 2003), magnetohydrodynamics (Knoll and Rider 1997), radiation transport (Shestakov and Milovich 2000), reacting flow (Smooke, Mitchell and Keyes 1989), structural analysis (Kant and Patel 1990) and circuit simulation (Grasser 1999) to overcome the problem with Newton's method we see in the example. Newton's method, even with a line search, can converge to non-physical solutions or unstable local minima of the norm of the steady-state residual (Keyes and Smooke 1987, Coffey *et al.* 2003*b*). This is particularly the case when the solution has complex features, such as shocks or discontinuities, that are not present in the initial iterate (Orkwis and McRae 1992).

We will express the method in terms of an initial value problem for (7.1):

$$\dot{\mathbf{x}} = -\mathbf{V}^{-1}\mathbf{F}(\mathbf{x}), \quad \mathbf{x}(0) = \mathbf{x}_0. \tag{7.3}$$

The matrix $\mathbf{V}$ plays the role of a scaling or preconditioning operator. We seek to find the stable steady-state solution of (7.3) if it exists. With this viewpoint, ΨTC is not a general-purpose nonlinear solver, but rather a tool for dynamics. If there is no stable steady-state solution, ΨTC may well diverge. We will, therefore, assume that a stable steady-state solution of (7.3) exists.

In this article we will focus on one version of the algorithm,

$$\mathbf{x}_{n+1} = \mathbf{x}_n - (\delta_n^{-1}\mathbf{V} + \mathbf{F}'(\mathbf{x}_n))^{-1}\mathbf{F}(\mathbf{x}_n), \tag{7.4}$$

and its inexact formulation

$$\begin{aligned} \mathbf{x}_{n+1} &= \mathbf{x}_n + \mathbf{s}_n, \\ \|(\delta_n^{-1}\mathbf{V} + \mathbf{F}'(\mathbf{x}_n))\mathbf{s}_n + \mathbf{F}(\mathbf{x}_n)\| &\le \eta\|\mathbf{F}(\mathbf{x}_n)\|. \end{aligned} \tag{7.5}$$

A typical choice for $\delta_n$, especially in aerodynamics (Keyes 1995, Orkwis and McRae 1992, Venkatakrishnan 1989), is the 'switched evolution relaxation' (SER) method (Mulder and Leer 1985),

$$\delta_n = \delta_{n-1}\|\mathbf{F}(\mathbf{x}_{n-1})\|/\|\mathbf{F}(\mathbf{x}_n)\| = \delta_0\|\mathbf{F}(\mathbf{x}_0)\|/\|\mathbf{F}(\mathbf{x}_n)\|. \tag{7.6}$$

It is often useful to bound $\delta_n$ from above (Coffey, Kelley and Keyes 2003*a*, Fowler and Kelley 2005, Kelley and Keyes 1998) and replace (7.6) with

$$\delta_n = \phi\left(\delta_{n-1}\frac{\|\mathbf{F}(\mathbf{x}_{n-1})\|}{\|\mathbf{F}(\mathbf{x}_n)\|}\right). \tag{7.7}$$

In (7.7),

$$\phi(\xi) = \begin{cases} \xi & \xi \le \xi_t, \\ \delta_{\max} & \xi > \xi_t, \end{cases} \tag{7.8}$$

where either $\xi_t = \delta_{\max}$ or $\xi_t < \infty$ and $\delta_{\max} = \infty$.

We present the theorem for ODE dynamics from Kelley and Keyes (1998) in detail. ΨTC has also been applied to problems with differential algebraic dynamics where some components of $\mathbf{x}$ are not differentiated in the continuous formulation (Landau and Lifschitz 1959, Ern, Giovangigli, Keyes and Smooke 1994) and also to problems with non-smooth dynamics that arise from the application of flux-limiters in computational fluid dynamics (Coffey *et al.* 2003*b*, Fowler and Kelley 2005). In Section 7.1 we will give an example of a problem with non-smooth differentiable algebraic dynamics.

We will use the formal assumptions from Kelley and Keyes (1998). The assumptions are technical. Simply put, they say that $\mathbf{x}^*$ is a stable steady-state solution and that the standard assumptions (Assumption 2.1) hold.

**Assumption 7.1.**

- The initial value problem (7.3) has a solution $\mathbf{x}(t)$ and
$$\lim_{t\to\infty} \mathbf{x}(t) = \mathbf{x}^*.$$
- $\mathbf{F}$ is Lipschitz continuously differentiable in the set
$$\mathcal{S} = \cup_{t\geq 0}\{\mathbf{x} \mid \|\mathbf{x} - \mathbf{x}(t)\| \leq \Delta\}$$
for some $\Delta > 0$.
- There is $M > 0$ such that $\|\mathbf{F}'(\mathbf{x})\| \leq M$ for all $\mathbf{x} \in \mathcal{S}$.
- There are $\epsilon$ and $\beta$ such that
$$\|(\mathbf{I} - \delta \mathbf{V}^{-1}\mathbf{F}'(\mathbf{x}))^{-1}\| \leq (1+\beta\delta)^{-1}$$
for all $\delta > 0$.

**Theorem 7.2.** Let Assumption 7.1 hold and let the update for $\delta$ be given by (7.7). Let $\{\mathbf{x}_n\}$ be the iteration (7.5). Then there are $\bar{\eta}$ and $\bar{\delta}$ such that if $\delta_0 \leq \bar{\delta}$ and $\eta_n \leq \bar{\eta}$ for all $n$, then $\mathbf{x}_n \to \mathbf{x}^*$ and $\delta_n \to \delta_{\max}$. Moreover, for $n$ sufficiently large,
$$\|\mathbf{e}_{n+1}\| = O((\eta_n + \delta_n^{-1})\|\mathbf{e}_n\| + \|\mathbf{e}_n\|^2).$$

### *7.1. Extensions to DAE and semi-smooth dynamics*

Coffey *et al.* (2003*a*, 2003*b*), Kelley *et al.* (2008), Farthing *et al.* (2003) and Fowler and Kelley (2005) extended the convergence results on ΨTC to the case of semi-explicit index-one differential algebraic equations (DAE):
$$\mathbf{D}\begin{pmatrix}\mathbf{u}\\ \mathbf{v}\end{pmatrix}' = -\begin{pmatrix}\mathbf{F}_1(\mathbf{u},\mathbf{v})\\ \mathbf{F}_2(\mathbf{u},\mathbf{v})\end{pmatrix} \equiv -\mathbf{F}(\mathbf{x}), \quad \mathbf{x}(0) = \mathbf{x}_0. \tag{7.9}$$

Here
$$\mathbf{x} = (\mathbf{u}^T, \mathbf{v}^T)^T \in C([0,\infty], \mathbb{R}^{N_1+N_2}).$$

The functions $\mathbf{u} : [0, \infty] \to \mathbb{R}^{N_1}$ and $\mathbf{v} : [0, \infty] \to \mathbb{R}^{N_2}$ are to be found. The differential variables $u$ and the algebraic variables $v$ are clearly separated in the semi-explicit case where

$$\mathbf{D} = \begin{pmatrix} \mathbf{D}_{11} & 0 \\ 0 & 0 \end{pmatrix},$$

where $\mathbf{D}_{11}$ is a non-singular scaling matrix. We assume that the Jacobian of $\mathbf{F}_2$ in $\mathbf{v}$ is non-singular (index one). A good general reference for DAEs is the book by Brenan, Campbell and Petzold (1996).

We assume the initial data for (7.9) are consistent (*i.e.* $\mathbf{F}_2(\mathbf{u}(0), \mathbf{v}(0)) = 0$), and seek the solution $\mathbf{x}^*$ to $\mathbf{F}(\mathbf{x}^*) = 0$ that satisfies

$$\lim_{t\to\infty} \mathbf{x}(t) = \mathbf{x}^*.$$

If (7.9) is a discretization in space of a PDE, and the initial data are far from the desired steady state, the application of a conventional method, such as a line search (Kelley 1995), to the time-independent equation

$$\mathbf{F}(\mathbf{x}) = 0$$

may fail to converge. Possible failure modes (Coffey *et al.* 2003*b*) are stagnation of the iteration at a singularity of $\mathbf{F}'$, the Jacobian of $\mathbf{F}$, or finding a solution other than $\mathbf{x}^*$.

The $\Psi$TC iteration for these dynamics in the smooth case is

$$\mathbf{x}_{n+1} = \mathbf{x}_n - (\delta_n^{-1}\mathbf{D} + \mathbf{F}'(\mathbf{x}_n))^{-1}\mathbf{F}(\mathbf{x}_n) \qquad (7.10)$$

(Coffey *et al.* 2003*a*). The difference is only that $\mathbf{D}$ is singular and $\mathbf{F}$ has the special structure of (7.9). The assumption that the DAE has index one is all one needs to obtain a convergence result exactly like Theorem 7.2. For the non-smooth case (Fowler and Kelley 2005), one must replace the Jacobians in the algorithm and in the definition of index one by the appropriate semi-smooth generalized derivatives.

### *7.2. Combustion application*

This example is taken from Chen (2001), Aziz, Stephens and Suri (1988), Barrett and Shanahan (1991) and Fowler and Kelley (2005). We globalize the semi-smooth Newton iteration with $\Psi$TC for a DAE. In the context of Section 7.1, $N_1 = N_2$.

We consider the boundary value problem

$$-u_{zz} + \lambda \max(0, u)^p = 0, \quad z \in (0, 1) \qquad (7.11)$$

(Aziz *et al.* 1988, Barrett and Shanahan 1991), with boundary data

$$u(0) = u(1) = 0 \qquad (7.12)$$

and $p \in (0, 1)$.

We reformulate the problem to make the forcing term Lipschitz continuous by adding a new variable,

$$v = \begin{cases} u^p & \text{if } u \geq 0, \\ u & \text{if } u < 0, \end{cases}$$

to obtain a Lipschitz continuous elliptic–algebraic system, $\mathcal{F}(w) = 0$, where $w = (u, v)^T$ and

$$\mathcal{F}(w) = \begin{pmatrix} f(u,v) \\ g(u,v) \end{pmatrix} = \begin{pmatrix} -u_{zz} + \lambda \max(0,v) \\ u - \omega(v) \end{pmatrix} = 0, \tag{7.13}$$

where

$$\omega(v) = \begin{cases} v^{1/p} & \text{if } v \geq 0, \\ v & \text{if } v < 0. \end{cases}$$

If we discretize the Laplacian with the standard central difference scheme with $N$ interior grid points, we obtain a finite-dimensional system $\mathbf{F}(\mathbf{w}) = 0$ for

$$\mathbf{w} = \begin{pmatrix} \mathbf{u} \\ \mathbf{v} \end{pmatrix} \in \mathbb{R}^{2N},$$

where

$$\mathbf{F}(\mathbf{w}) = \begin{pmatrix} \mathbf{F}_1(\mathbf{u},\mathbf{v}) \\ \mathbf{F}_2(\mathbf{u},\mathbf{v}) \end{pmatrix} = \begin{pmatrix} -\mathbf{L}_{\delta_z}\mathbf{u} \\ \mathbf{u} - \mathbf{v} - \max(0,\mathbf{v})^{1/p} \end{pmatrix} + \begin{pmatrix} \lambda \\ 1 \end{pmatrix} \max(0,\mathbf{v}).$$

Here functions are understood to be componentwise evaluations, $\mathbf{L}_{\delta_z}$ is the discretized Laplacian, and $\delta_z$ is the spatial mesh width.

The reason we formulate the problem with DAE (rather than ODE) dynamics is that the pseudo-time variable should not be added to both equations in (7.13) but only the first. The reason for this is that the true time-dependent system is

$$u_t = u_{zz} - \lambda \max(0,u)^p,$$

and that the auxiliary variable $v$ is used only to make the nonlinearity Lipschitz continuous. One might think that an ODE formulation would work equally well, but in fact the ODE formulation, which does not model the physics, failed to converge in our testing.

ΨTC for this problem, which is semi-smooth, looks like (7.10) with $\mathbf{F}'$ replaced by $\mathbf{V} \in \partial\mathbf{F}(\mathbf{w})$:

$$\mathbf{w}_{n+1} = \mathbf{w}_n - (\delta_n^{-1}\mathbf{D} + \mathbf{V}(\mathbf{w}_n))^{-1}\mathbf{F}(\mathbf{w}_n), \tag{7.14}$$

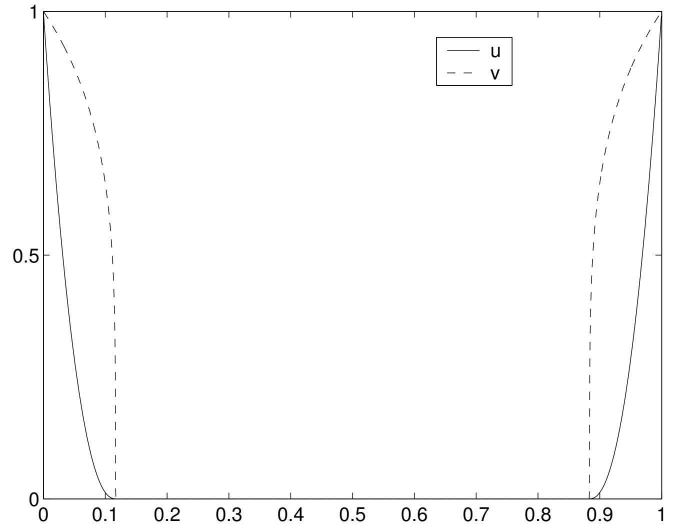

Figure 7.1. Solution to (7.11) and (7.12) via (7.14) for $p = 0.1$, $\lambda = 200$.

where $\mathbf{D}_{11} = \mathbf{I}$. One can compute $\partial \mathbf{F}$ analytically using the well-known result for the scalar function $\max(0, v)$

$$\partial \max(0, v) = \begin{cases} 0 & \text{if } v < 0, \\ [0, 1] & \text{if } v = 0, \\ 1 & \text{if } v > 0. \end{cases}$$

Hence

$$\partial \mathbf{F} = \begin{pmatrix} -L_{\delta_z} & 0 \\ \mathbf{I} & -1 - (1/p) \max(0, \mathbf{V})^{(1-p)/p} \end{pmatrix} + \begin{pmatrix} 0 & \lambda \mathbf{I} \\ 0 & \mathbf{I} \end{pmatrix} \partial \max(0, \mathbf{V}). \tag{7.15}$$

Here $\max(0, \mathbf{F})$ and $\partial \max(0, \mathbf{V})$ denote the diagonal matrices with the entries of the vector on the diagonal.

We report on one of the computations in Fowler and Kelley (2005) with $p = 0.1$ and $\lambda = 200$. This choice leads to a large 'dead core' (Aziz *et al.* 1988, Barrett and Shanahan 1991), a region in which the solution vanishes. We plot the solution in Figure 7.1.

In the continuation we could use any choice from the set-valued map $\partial \max(0, v)$, and selected

$$\chi(v) = \left\{ \begin{matrix} 0 & \text{if } v \le 0 \\ 1 & \text{if } v > 0 \end{matrix} \right\} \in \partial \max(0, v).$$

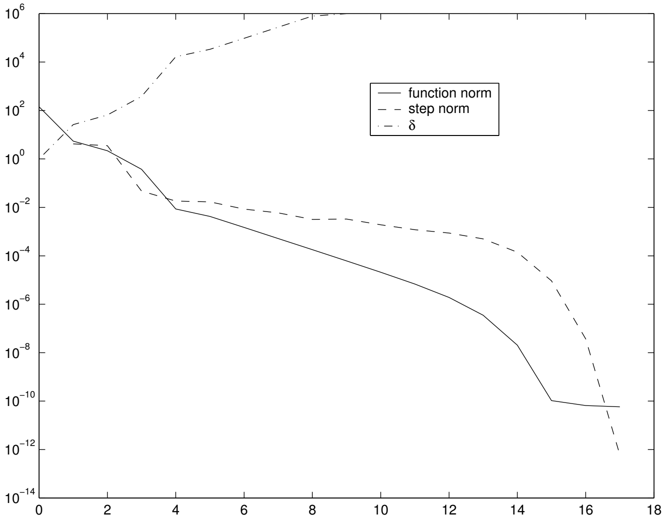

Figure 7.2. Residual history: semi-smooth $\Psi$TC.

We used $\delta_0 = 1$ and $\delta_{\max} = 10^6$. We terminate the nonlinear iteration when either

$$\|\mathbf{F}(\mathbf{w}_n)\|/\|\mathbf{F}(\mathbf{w}_0)\| < 10^{-13} \quad \text{or} \quad \|\mathbf{s}_n\| < 10^{-10}, \tag{7.16}$$

where $\mathbf{s}_n = \mathbf{w}_{n+1} - \mathbf{w}_n$. In the tables we see the superlinear convergence clearly in the reduction in the norms of the steps; this is consistent with the estimate $\mathbf{s}_n = -\mathbf{e}_n + o(\|\mathbf{e}_n\|)$ which follows from local superlinear convergence. The superlinear convergence is less visible in the residual norms, because the generalized Jacobians become more ill-conditioned as the mesh is refined. The residual norms begin to stagnate after a reduction of $10^{12}$.

In Figure 7.2, taken from Fowler and Kelley (2005), we plot the norms of the steps and nonlinear residuals together with the growth of $\delta$ for a mesh of width $\delta_z = 1/2048$. $\delta$ grows smoothly in the early phase of the iteration and reaches its maximum rapidly. The superlinear convergence is clearly visible in the curve for the norms of the steps. The Jacobian of the nonlinear residual has a condition number of $O(1/h^2)$, and hence the residual norm reflects the error less accurately.

## Acknowledgements

The author's work on this paper has been partially supported by the Consortium for Advanced Simulation of Light Water Reactors (www.casl.gov), an Energy Innovation Hub (www.energy.gov/hubs) for Modeling and Simulation of Nuclear Reactors under the US Department of Energy Contract no. DE-AC05-00OR22725, Army Research Office grant W911NF-16-1-0504, National Science Foundation grants ACI-1740309, DMS-1406349, and

National Science Foundation grant DMS-1638521 to the Statistical and Applied Mathematical Sciences Institute. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the Army Research Office, the Department of Energy or the National Science Foundation.

## REFERENCES[2]

P.-A. Absil, C. G. Baker and K. A. Gallivan (2007), 'Trust-region methods on Riemannian manifolds', *Found. Comput. Math.* **7**, 303–330.

E. L. Allgower, K. Böhmer, F. A. Potra and W. C. Rheinboldt (1986), 'A mesh-independence principle for operator equations and their discretizations', *SIAM J. Numer. Anal.* **23**, 160–169.

H. An, X. Jia and H. F. Walker (2017), 'Anderson acceleration and application to the three-temperature energy equations', *J. Comput. Phys.* **347**, 1–19.

D. G. Anderson (1965), 'Iterative procedures for nonlinear integral equations', *J. Assoc. Comput. Mach.* **12**, 547–560.

L. Armijo (1966), 'Minimization of functions having Lipschitz-continuous first partial derivatives', *Pacific J. Math.* **16**, 1–3.

U. M. Ascher and L. R. Petzold (1998), *Computer Methods for Ordinary Differential Equations and Differential Algebraic Equations*, SIAM.

U. M. Ascher, R. M. M. Mattheij and R. D. Russell (1995), *Numerical Solution of Boundary Value Problems for Ordinary Differential Equations*, Classics in Applied Mathematics, SIAM.

A. K. Aziz, A. B. Stephens and M. Suri (1988), 'Numerical methods for reaction–diffusion problems with non-differentiable kinetics', *Numer. Math.* **53**, 1–11.

S. Balay, S. Abhyankar, M. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, V. Eijkhout, W. Gropp, D. Kaushik, M. Knepley, L. C. McInnes, K. Rupp, B. Smith, S. Zampini and H. Zhang (2015), PETSc Users Manual, revision 3.6. Technical report ANL-95/11 Rev 3.6, Mathematics and Computer Science Division, Argonne National Laboratory.

J. W. Barrett and R. M. Shanahan (1991), 'Finite element approximation of a model reaction–diffusion problem with a non-Lipschitzian nonlinearity', *Numer. Math.* **59**, 217–242.

G. Bratu (1914), 'Sur les équations intégrales non linéaires', *Bull. Math. Soc. France* **42**, 113–142.

K. E. Brenan, S. L. Campbell and L. R. Petzold (1996), *The Numerical Solution of Initial Value Problems in Differential-Algebraic Equations*, Vol. 14 of Classics in Applied Mathematics, SIAM.

R. Brent (1973), 'Some efficient algorithms for solving systems of nonlinear equations', *SIAM J. Numer. Anal.* **10**, 327–344.

E. L. Briggs, D. J. Sullivan and J. Bernholc (1995), 'Large-scale electronic-structure calculations with multigrid acceleration', *Phys. Rev. B* **52**, R5471–R5474.

---

[2] The URLs cited in this work were correct at the time of going to press, but the publisher and the authors make no undertaking that the citations remain live or are accurate or appropriate.

W. Burmeister (1975), Zur Konvergenz einiger verfahren der konjugierten Richtungen. In *Internationaler Kongreß über Anwendung der Mathematik in dem Ingenieurwissenschaften*, Weimar.

I. W. Busbridge (1960), *The Mathematics of Radiative Transfer*, Vol. 50 of Cambridge Tracts in Mathematics and Mathematical Physics, Cambridge University Press.

X.-C. Cai, W. D. Gropp, D. E. Keyes and M. D. Tidriri (1994), Newton–Krylov–Schwarz methods in CFD. In *Proceedings of the International Workshop on the Navier–Stokes Equations* (R. Rannacher, ed.), Notes in Numerical Fluid Mechanics, Vieweg.

S. L. Campbell, I. C. F. Ipsen, C. T. Kelley and C. D. Meyer (1996*a*), 'GMRES and the minimal polynomial', *BIT* **36**, 664–675.

S. L. Campbell, I. C. F. Ipsen, C. T. Kelley, C. D. Meyer and Z. Q. Xue (1996*b*), 'Convergence estimates for solution of integral equations with GMRES', *J. Integral Equ. Appl.* **8**, 19–34.

N. N. Carlson and K. Miller (1998), 'Design and application of a gradient weighted moving finite element code I: In one dimension', *SIAM J. Sci. Comput.* **19**, 766–798.

S. Chandrasekhar (1960), *Radiative Transfer*, Dover.

X. Chen (2000), 'Smoothing methods for complementarity problems and their applications: A survey', *J. Oper. Res. Soc. Japan* **43**, 32–47.

X. Chen (2001), 'A superlinearly and globally convergent method for reaction and diffusion problems with a non-Lipschitzian operator', *Computing Supplementum* **15**, 79–90.

X. Chen and C. T. Kelley (2017), Analysis of the EDIIS algorithm. Preprint.

X. Chen and T. Yamamoto (1989), 'Convergence domains of certain iterative methods for solving nonlinear equations', *Numer. Funct. Anal. Optim.* **10**, 37–48.

X. Chen and Y. Ye (1999), 'On homotopy-smoothing methods for box-constrained variational inequalities', *SIAM J. Control Optim.* **37**, 589–616.

X. Chen, Z. Nashed and L. Qi (2001), 'Smoothing methods and semismooth methods for nondifferentiable operator equations', *SIAM J. Numer. Anal.* **38**, 1200–1216.

X. Chen, L. Qi and D. Sun (1998), 'Global and superlinear convergence of the smoothing Newton method and its application to general box constrained variational inequalities', *Math. Comp.* **67**, 519–540.

F. H. Clarke (1990), *Optimization and Nonsmooth Analysis*, Vol. 5 of Classics in Applied Mathematics, SIAM.

T. Coffey, C. T. Kelley and D. E. Keyes (2003*a*), 'Pseudo-transient continuation and differential-algebraic equations', *SIAM J. Sci. Comput.* **25**, 553–569.

T. S. Coffey, R. J. McMullan, C. T. Kelley and D. S. McRae (2003*b*), 'Globally convergent algorithms for nonsmooth nonlinear equations in computational fluid dynamics', *J. Comput. Appl. Math.* **152**, 69–81.

A. M. Collier, A. C. Hindmarsh, R. Serban and C. S. Woodward (2015), User documentation for KINSOL v2.8.0. Technical report UCRL-SM-208116, Lawrence Livermore National Laboratory.

A. R. Conn, N. I. M. Gould and P. L. Toint (2000), *Trust Region Methods*, Vol. 1 of MPS–SIAM Series on Optimization, SIAM.

W. M. Coughran and J. W. Jerome (1990), Modular algorithms for transient semiconductor device simulation I: Analysis of the outer iteration. In *AMS-SIAM Summer Seminar on Device Simulation* (R. E. Bank, ed.), Vol. 25 of AMS Lectures in Applied Mathematics, AMS, pp. 107–149.

M. G. Crandall and P. H. Rabinowitz (1971), 'Bifurcation from simple eigenvalues', *J. Funct. Anal.* **8**, 321–340.

A. R. Curtis, M. J. D. Powell and J. K. Reid (1974), 'On the estimation of sparse Jacobian matrices', *J. Inst. Math. Appl.* **13**, 117–119.

CVX Research, Inc. (2012), CVX: Matlab software for disciplined convex programming, version 2.0. http://cvxr.com/cvx

D. W. Decker and C. T. Kelley (1980), 'Newton's method at singular points I', *SIAM J. Numer. Anal.* **17**, 66–70.

D. W. Decker and C. T. Kelley (1983), 'Sublinear convergence of the chord method at singular points', *Numer. Math.* **42**, 147–154.

T. De Luca, F. Facchinei and C. Kanzow (1996), 'A semismooth equation approach to the solution of nonlinear complementarity problems', *Math. Program.* **75**, 407–439.

R. Dembo, S. Eisenstat and T. Steihaug (1982), 'Inexact Newton methods', *SIAM J. Numer. Anal.* **19**, 400–408.

J. W. Demmel (1997), *Applied Numerical Linear Algebra*, SIAM.

J. E. Dennis (1969), 'On the Kantorovich hypothesis for Newton's method', *SIAM J. Numer. Anal.* **6**, 493–507.

J. E. Dennis (1971), Toward a unified convergence theory for Newton-like methods. In *Nonlinear Functional Analysis and Applications* (L. B. Rall, ed.), Academic, pp. 425–472.

J. E. Dennis and R. B. Schnabel (1979), 'Least change secant updates for quasi-Newton methods', *SIAM Review* **21**, 443–459.

J. E. Dennis and R. B. Schnabel (1996), *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Vol. 16 of Classics in Applied Mathematics, SIAM.

J. E. Dennis and H. F. Walker (1981), 'Convergence theorems for least change secant update methods', *SIAM J. Numer. Anal.* **18**, 949–987.

P. Deuflhard (2004), *Newton Methods for Nonlinear Problems: Affine Invariance and Adaptive Algorithms*, Vol. 35 of Computational Mathematics, Springer.

P. Deuflhard, R. W. Freund and A. Walter (1990), 'Fast secant methods for the iterative solution of large nonsymmetric linear systems', *Impact Comput. Sci. Engrg* **2**, 244–276.

E. J. Doedel (1997), *Lecture Notes on Numerical Analysis of Bifurcation Problems*, from Sommerschule über Nichtlineare Gleichungssysteme, Hamburg, Germany, March 17–21, 1997. Available by anonymous ftp to: ftp.cs.condordia.ca in pub/doedel/doc/hamburg.ps.Z

E. J. Doedel and J. P. Kernévez (1986), AUTO: Software for continuation and bifurcation problems in ordinary differential equations. Technical report, California Institute of Technology.

S. C. Eisenstat and H. F. Walker (1996), 'Choosing the forcing terms in an inexact Newton method', *SIAM J. Sci. Comput.* **17**, 16–32.

A. Ern, V. Giovangigli, D. E. Keyes and M. D. Smooke (1994), 'Towards polyalgorithmic linear system solvers for nonlinear elliptic problems', *SIAM J. Sci. Comput.* **15**, 681–703.

F. Facchinei, A. Fischer and C. Kanzow (1996), Inexact Newton methods for semismooth equations with applications to variational inequality problems. In *Nonlinear Optimization and Applications* (G. D. Pillo and F. Giannessi, eds), Plenum, pp. 125–139.

H.-R. Fang and Y. Saad (2009), 'Two classes of multisecant methods for nonlinear acceleration', *Numer. Linear Algebra Appl.* **16**, 197–221.

M. W. Farthing, C. E. Kees, T. Coffey, C. T. Kelley and C. T. Miller (2003), 'Efficient steady-state solution techniques for variably saturated groundwater flow', *Adv. Water Resour.* **26**, 833–849.

H. Federer (1969), *Geometric Measure Theory*, Vol. 153 of Grundlehren der mathematischen Wissenschaften, Springer.

W. R. Ferng and C. T. Kelley (2000), 'Mesh independence of matrix-free methods for path following', *SIAM J. Sci. Comput.* **21**, 1835–1850.

A. Fischer (1992), 'A special Newton-type optimization method', *Optimization* **24**, 269–284.

J. B. Foresman and A. Frisch (1996), *Exploring Chemistry with Electronic Structure Methods*, second edition, Gaussian, Inc.

K. R. Fowler and C. T. Kelley (2005), 'Pseudo-transient continuation for nonsmooth nonlinear equations', *SIAM J. Numer. Anal.* **43**, 1385–1406.

M. J. Frisch, G. W. Trucks, H. B. Schlegel, G. E. Scuseria, M. A. Robb, J. R. Cheeseman, G. Scalmani, V. Barone, B. Mennucci, G. A. Petersson, H. Nakatsuji, W. Caricato, X. Li, H. P. Hratchian, A. F. Izmaylov, J. Bloino, G. Zheng, J. L. Sonnenberg, M. Hada, M. Ehara, K. Toyota, R. Fukuda, J. Hasegawa, M. Ishida, T. Nakajima, Y. Honda, D. Kitao, H. Nakai, T. Vreven, J. A. Montgomery Jr, J. E. Peralta, F. Ogliaro, M. Bearpark, J. J. Heyd, E. Brothers, K. N. Kudin, V. N. Staroverov, R. Kobayashi, J. Normand, K. Raghavachari, A. Rendell, J. C. Burant, S. S. Iyengar, J. Tomasi, M. Cossi, N. Rega, J. M. Millam, M. Klene, J. E. Knox, J. B. Cross, V. Bakken, C. Adamo, J. Jaramillo, R. Gomperts, R. E. Stratmann, O. Yazyev, A. J. Austin, R. Cammi, C. Pomelli, J. W. Ochterski, R. L. Martin, K. Morokuma, V. G. Zakrzewski, G. A. Voth, P. Salvador, J. J. Dannenberg, S. Dapprich, A. D. Daniels, Ö. Farkas, J. B. Foresman, J. V. Ortiz, J. Cioslowski and D. J. Fox (2009), *Gaussian 09, Revision A.1*, Gaussian, Inc.

V. Ganine, U. Javiya, N. Hills and J. Chew (2012), 'Coupled fluid–structure transient thermal analysis of a gas turbine internal air system with multiple cavities', *J. Engrg Gas Turbines Power* **134**, 102508.

D. J. Gardner, C. S. Woodward, D. R. Reynolds, G. Hommes, S. Aubrey and A. Arsnelis (2015), 'Implicit integration methods for dislocation dynamics', *Modelling Simul. Mater. Sci. Engrg* **23**, 025006.

G. H. Golub and M. A. Saunders (1969), Linear least squares and quadratic programming. Technical report CS 134, Stanford University.

G. H. Golub and C. G. Van Loan (1996), *Matrix Computations*, third edition, Johns Hopkins University Press.

W. J. F. Govaerts (2000), *Numerical Methods for Bifurcations of Dynamic Equilibria*, SIAM.

K.-T. Grasser (1999), Mixed-mode device simulation. Technical report, Technical University of Vienna (doctoral dissertation).
http://www.iue.tuwien.ac.at/phd/grasser/

S. Hamilton, M. Berrill, K. Clarno, R. Pawlowski, A. Toth, C. T. Kelley, T. Evans and B. Philip (2016), 'An assessment of coupling algorithms for nuclear reactor core physics simulations', *J. Comput. Phys.* **311**, 241–257.

W. E. Hart and S. O. W. Soul (1973), 'Quasi-Newton methods for discretized nonlinear boundary problems', *J. Inst. Appl. Math.* **11**, 351–359.

M. Heinkenschloß, C. T. Kelley and H. T. Tran (1992), 'Fast algorithms for nonsmooth compact fixed point problems', *SIAM J. Numer. Anal.* **29**, 1769–1792.

M. Heinkenschloss, M. Ulbrich and S. Ulbrich (1999), 'Superlinear and quadratic convergence of affine scaling interior-point Newton methods for problems with simple bounds and without strict complementarity assumption', *Math. Program.* **86**, 615–635.

M. A. Heroux, R. A. Bartlett, V. E. Howle, R. J. Hoekstra, J. J. Hu, T. G. Kolda, R. B. Lehoucq, K. R. Long, R. P. Pawlowski, E. T. Phipps, A. G. Salinger, H. K. Thornquist, R. S. Tuminaro, J. M. Willenbring, A. Williams and K. S. Stanley (2005), An overview of the Trilinos project. Technical report 3, Sandia National Laboratories.

D. J. Higham (1999), 'Trust region algorithms and time step selection', *SIAM J. Numer. Anal.* **37**, 194–210.

N. J. Higham (1996), *Accuracy and Stability of Numerical Algorithms*, SIAM.

A. C. Hindmarsh, P. N. Brown, K. E. Grant, S. L. Lee, R. Serban, D. E. Shumaker and C. S. Woodward (2005), 'SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers', *ACM Trans. Math. Softw.* **31**, 363–396.

M. Hintermüller (2010), Semismooth Newton methods and applications. Oberwolfach Seminar on 'Mathematics of PDE-Constrained Optimization' at Mathematisches Forschungsinstitut in Oberwolfach, November 2010.

M. Hintermüller and M. Ulbrich (2003), A mesh-independence result for semismooth Newton methods. Technical report, Fachbereich Mathematik, Universität Hamburg.

P. Hohenberg and W. Kohn (1964), 'Inhomogeneous electron gas', *Phys. Rev.* **136**, B864–B871.

H. Jiang and L. Qi (1997), 'A new nonsmooth equations approach to nonlinear complementarity problems', *SIAM J. Control Optim.* **35**, 178–193.

T. Kant and S. Patel (1990), 'Transient/pseudo-transient finite element small/large deformation analysis of two-dimensional problems', *Comput. Structures* **36**, 421–427.

L. Kantorovich and G. Akilov (1982), *Functional Analysis*, second edition, Pergamon.

S. Karlin (1959), 'Positive operators', *J. Math. Mech.* **8**, 907–937.

H. B. Keller (1987), *Lectures on Numerical Methods in Bifurcation Theory*, Tata Institute of Fundamental Research, Lectures on Mathematics and Physics, Springer.

C. T. Kelley (1994), Identification of the support of nonsmoothness. In *Large Scale Optimization: State of the Art* (W. W. Hager, D. W. Hearn and P. Pardalos, eds), Kluwer Academic, pp. 192–205.

C. T. Kelley (1995), *Iterative Methods for Linear and Nonlinear Equations*, Vol. 16 of Frontiers in Applied Mathematics, SIAM.

C. T. Kelley (1999), *Iterative Methods for Optimization*, Vol. 18 of Frontiers in Applied Mathematics, SIAM.

C. T. Kelley and D. E. Keyes (1998), 'Convergence analysis of pseudo-transient continuation', *SIAM J. Numer. Anal.* **35**, 508–523.

C. T. Kelley and E. W. Sachs (1985), 'Broyden's method for approximate solution of nonlinear integral equations', *J. Integral Equations* **9**, 25–44.

C. T. Kelley and E. W. Sachs (1987), 'A quasi-Newton method for elliptic boundary value problems', *SIAM J. Numer. Anal.* **24**, 516–531.

C. T. Kelley and E. W. Sachs (1989), 'A pointwise quasi-Newton method for unconstrained optimal control problems', *Numer. Math.* **55**, 159–176.

C. T. Kelley and E. W. Sachs (1991), 'Mesh independence of Newton-like methods for infinite dimensional problems', *J. Integral Equ. Appl.* **3**, 549–573.

C. T. Kelley and E. W. Sachs (1993), 'Pointwise Broyden methods', *SIAM J. Optim.* **3**, 423–441.

C. T. Kelley and E. W. Sachs (1994), 'Multilevel algorithms for constrained compact fixed point problems', *SIAM J. Sci. Comput.* **15**, 645–667.

C. T. Kelley and E. W. Sachs (1995), 'Solution of optimal control problems by a pointwise projected Newton method', *SIAM J. Control Optim.* **33**, 1731–1757.

C. T. Kelley and Z. Q. Xue (1996), 'GMRES and integral operators', *SIAM J. Sci. Comput.* **17**, 217–226.

C. T. Kelley, L.-Z. Liao, L. Qi, M. T. Chu, J. P. Reese and C. Winton (2008), 'Projected pseudo-transient continuation', *SIAM J. Numer. Anal.* **46**, 3071–3083.

T. Kerkhoven and J. W. Jerome (1990), '$L_\infty$ stability of finite element approximations to elliptic gradient equations', *Numer. Math.* **57**, 561–575.

D. E. Keyes (1995), Aerodynamic applications of Newton–Krylov–Schwarz solvers. In *Proceedings of the 14th International Conference on Numerical Methods in Fluid Dynamics* (R. Narasimha, ed.), Springer, pp. 1–20.

D. E. Keyes and M. D. Smooke (1987), 'Flame sheet starting estimates for counterflow diffusion flame problems', *J. Comput. Phys.* **72**, 267–288.

D. A. Knoll and D. E. Keyes (2004), 'Jacobian-free Newton–Krylov methods: A survey of approaches and applications', *J. Comput. Phys.* **193**, 357–397.

D. A. Knoll and W. J. Rider (1997), A multigrid preconditioned Newton–Krylov method. Technical report LA-UR-97-4013, Los Alamos National Laboratory.

D. A. Knoll, H. Park and K. Smith (2011), 'Application of the Jacobian-free Newton–Krylov method to nonlinear acceleration of transport source iteration in slab geometry', *Nuclear Sci. Engrg* **167**, 122–132.

W. Kohn and L. J. Sham (1965), 'Self-consistent equations including exchange and correlation effects', *Phys. Rev.* **140**, A1133–A1138.

K. N. Kudin, G. E. Scuseria and E. Cancès (2002), 'A black-box self-consistent field convergence algorithm: One step closer', *J. Chem. Phys.* **116**, 8255–8261.

Y. A. Kuznetsov (1998), *Elements of Applied Bifurcation Theory*, Springer.

L. D. Landau and E. M. Lifschitz (1959), *Fluid Mechanics*, Pergamon.

R. J. LeVeque (2007), *Finite Difference Methods for Ordinary and Partial Differential Equations*, SIAM.

L. Lin and C. Yang (2013), 'Elliptic preconditioner for accelerating the self-consistent field iteration in Kohn–Sham density functional theory', *SIAM J. Sci. Comput.* **35**, S277–S298.

P. A. Lott, H. F. Walker, C. S. Woodward and U. M. Yang (2012), 'An accelerated Picard method for nonlinear systems related to variably saturated flow', *Adv. Water Resour.* **38**, 92–101.

J. E. Marsden and M. McCracken (1976), *The Hopf Bifurcation and its Applications*, Vol. 19 of Applied Mathematical Sciences, Springer.

J. Martinez and L. Qi (1995), 'Inexact Newton methods for solving nonsmooth equations', *J. Comput. Appl. Math.* **60**, 127–145.

R. Mifflin (1977), 'Semismooth and semiconvex functions in constrained optimization', *SIAM J. Control Optim.* **15**, 959–972.

K. Miller (2005), 'Nonlinear Krylov and moving nodes in the method of lines', *J. Comput. Appl. Math.* **183**, 275–287.

J. J. Moré and G. Toraldo (1991), 'On the solution of large quadratic programming problems with bound constraints', *SIAM J. Optim.* **1**, 93–113.

W. Mulder and B. V. Leer (1985), 'Experiments with implicit upwind methods for the Euler equations', *J. Comput. Phys.* **59**, 232–246.

A. Neumaier (1998), MINQ: General definite and bound constrained indefinite quadratic programming. http://www.mat.univie.ac.at/~neum/software/minq/

O. Nevanlinna (1993), *Convergence of Iterations for Linear Equations*, Birkhäuser.

I. Newton (1967–1976), *The Mathematical Papers of Isaac Newton* (seven volumes, D. T. Whiteside, ed.), Cambridge University Press.

J. Nocedal and S. J. Wright (1999), *Numerical Optimization*, Springer.

C. W. Oosterlee and T. Washio (2000), 'Krylov subspace acceleration for nonlinear multigrid schemes', *SIAM J. Sci. Comput.* **21**, 1670–1690.

P. D. Orkwis and D. S. McRae (1992), 'Newton's method solver for the axisymmetric Navier–Stokes equations', *AIAA J.* **30**, 1507–1514.

J. M. Ortega and W. C. Rheinboldt (1970), *Iterative Solution of Nonlinear Equations in Several Variables*, Academic.

M. L. Overton (2001), *Numerical Computing with IEEE Floating Point Arithmetic*, SIAM.

J. S. Pang and L. Qi (1993), 'Nonsmooth equations: Motivation and algorithms', *SIAM J. Optim.* **3**, 443–645.

L. R. Petzold (1983), A description of DASSL: A differential/algebraic system solver. In *Scientific Computing* (R. S. Stepleman *et al.*, eds), North-Holland, pp. 65–68.

E. Picard (1890), 'Mémoire sur la théorie des équations aux dérivées partielles et la méthode des approximations successives', *J. de Math. ser. 4* **6**, 145–210.

F. A. Potra and H. Engler (2013), 'A characterization of the behavior of the Anderson acceleration on linear problems', *Linear Algebra Appl.* **438**, 1002–1011.

M. J. D. Powell (1970), A hybrid method for nonlinear equations. In *Numerical Methods for Nonlinear Algebraic Equations* (P. Rabinowitz, ed.), Gordon & Breach, pp. 87–114.

P. Pulay (1980), ‘Convergence acceleration of iterative sequences. The case of SCF iteration’, *Chem. Phys. Lett.* **73**, 393–398.

P. Pulay (1982), ‘Improved SCF convergence acceleration’, *J. Comput. Chem.* **3**, 556–560.

L. Qi and J. Sun (1993), ‘A nonsmooth version of Newton’s method’, *Math. Program.* **58**, 353–367.

P. H. Rabinowitz (1971), ‘Some global results for nonlinear eigenvalue problems’, *J. Funct. Anal.* **7**, 487–513.

J. Raphson (1690), Analysis aequationum universalis seu ad aequationes algebraicas resolvendas methodus generalis, et expedita, ex nova infinitarum serierum doctrina, deducta ac demonstrata. Original in British Library, London.

W. C. Rheinboldt (1986), *Numerical Analysis of Parametrized Nonlinear Equations*, Wiley.

T. Rohwedder and R. Schneider (2011), ‘An analysis for the DIIS acceleration method used in quantum chemistry calculations’, *J. Math. Chem.* **49**, 1889–1914.

Y. Saad, J. R. Chelikowsky and S. M. Shontz (2010), ‘Numerical methods for electronic structure calculations of materials’, *SIAM Review* **52**, 3–54.

E. W. Sachs (1990), ‘Convergence of algorithms for perturbed optimization problems’, *Ann. Oper. Res.* **27**, 311–342.

A. G. Salinger, N. M. Bou-Rabee, R. P. Pawlowski, E. D. Wilkes, E. A. Burroughs, R. B. Lehoucq and L. A. Romero (2002), LOCA 1.0 Library of Continuation Algorithms: Theory and Implementation Manual. Technical report SAND2002-0396, Sandia National Laboratory.

R. Schneider, T. Rohwedder, A. Neelov and J. Blauert (2008), ‘Direct minimization for calculating invariant subspaces in density functional computations of the electronic structure’, *J. Comput. Math.* **27**, 360–387.

V. E. Shamanskii (1967), ‘A modification of Newton’s method’ (in Russian), *Ukran. Mat. Zh.* **19**, 133–138.

A. I. Shestakov and J. L. Milovich (2000), Applications of pseudo-transient continuation and Newton–Krylov methods for the Poisson–Boltzmann and radiation diffusion equations. Technical report UCRL-JC-139339, Lawrence Livermore National Laboratory.

V. Simoncini and D. B. Szyld (2003*a*), ‘Flexible inner–outer Krylov subspace methods’, *SIAM J. Numer. Anal.* **40**, 2219–2239.

V. Simoncini and D. B. Szyld (2003*b*), ‘Theory of inexact Krylov subspace methods and applications to scientific computing’, *SIAM J. Sci. Comput.* **25**, 454–477.

V. Simoncini and D. B. Szyld (2007), ‘Recent computational developments in Krylov subspace methods for linear systems’, *Numer. Linear Algebra with Appl.* **14**, 1–59.

M. D. Smooke, R. Mitchell and D. Keyes (1989), ‘Numerical solution of two-dimensional axisymmetric laminar diffusion flames’, *Combust. Sci. Tech.* **67**, 85–122.

R. A. Tapia, J. E. Dennis and J. P. Schäfermeyer (2018), ‘Inverse, shifted inverse, and Rayleigh quotient iteration as Newton’s method’. *SIAM Rev.* **60**, 3–55.

M. D. Tocci, C. T. Kelley and C. T. Miller (1997), 'Accurate and economical solution of the pressure head form of Richards' equation by the method of lines', *Adv. Water Resour.* **20**, 1–14.

A. Toth (2016), A theoretical analysis of Anderson acceleration and its application in multiphysics simulation for light-water reactors. PhD thesis, North Carolina State University.

A. Toth and C. T. Kelley (2015), 'Convergence analysis for Anderson acceleration', *SIAM J. Numer. Anal.* **53**, 805–819.

A. Toth and R. Pawlowski (2015), NOX::Solver::AndersonAcceleration Class Reference.
https://trilinos.org/docs/dev/packages/nox/doc/html/classNOX_1_1Solver_1_1AndersonAcceleration.html

A. Toth, J. A. Ellis, T. Evans, S. Hamilton, C. T. Kelley, R. Pawlowski and S. Slattery (2017), 'Local improvement results for Anderson acceleration with inaccurate function evaluations', **39**, S47–S65.

A. Toth, C. T. Kelley, S. Slattery, S. Hamilton, K. Clarno and R. Pawlowski (2015), Analysis of Anderson acceleration on a simplified neutronics/thermal hydraulics system. Joint International Conference on 'Mathematics and Computation (M&C), Supercomputing in Nuclear Applications (SNA) and the Monte Carlo (MC) Method'.

J. F. Traub (1964), *Iterative Methods for the Solution of Equations*, Prentice Hall.

M. Ulbrich (2001), 'Nonmonotone trust-region methods for bound-constrained semismooth equations with applications to nonlinear mixed complementarity problems', *SIAM J. Optim.* **11**, 889–917.

M. Ulbrich (2011), *Semismooth Newton Methods for Variational Inequalities and Constrained Optimization Problems in Function Spaces*, MOS-SIAM Series on Optimization, SIAM.

V. Venkatakrishnan (1989), 'Newton solution of inviscid and viscous problems', *AIAA J.* **27**, 885–891.

H. W. Walker and P. Ni (2011), 'Anderson acceleration for fixed-point iterations', *SIAM J. Numer. Anal.* **49**, 1715–1735.

T. Washio and C. Oosterlee (1997), 'Krylov subspace acceleration for nonlinear multigrid schemes', *Electron. Trans. Numer. Anal.* **6**, 271–290.

J. Willert, X. Chen and C. T. Kelley (2015), 'Newton's method for Monte Carlo-based residuals', *SIAM J. Numer. Anal.* **53**, 1738–1757.

J. Willert, C. T. Kelley, D. A. Knoll and H. K. Park (2013), 'Hybrid deterministic/Monte Carlo neutronics', *SIAM J. Sci. Comput.* **35**, S62–S83.

J. Willert, W. T. Taitano and D. Knoll (2014), 'Leveraging Anderson Acceleration for improved convergence of iterative solutions to transport systems', *J. Comput. Phys.* **273**, 278–286.

Y. Yang and L. Qi (2005), 'Smoothing trust region methods for nonlinear complementarity problems with $p_0$-functions', *Ann. Oper. Res.* **133**, 99–117.